solartron instruments

# 7081
# Computing Multimeter

MP MASTER
MA20080

## OPERATING MANUAL: PART 2

solartron
instruments

# 7081
# Precision Digital Voltmeter

## *OPERATING MANUAL*
## *Part 2*

Solartron Instruments is a division of the Solartron Group Ltd. Any reference to Schlumberger or Schlumberger Instruments (the name of the company until November 1993) implies no liability on the part of Schlumberger.

**Solartron Instruments**
**a division of the Solartron Group Ltd**
Victoria Road, Farnborough,
Hampshire GU14 7PW. UK.

Issue 5 : June 1985
Solartron Part No. : 70810011

19©94

**Solartron Instruments**
Victoria Road, Farnborough
Hampshire GU14 7PW England.
Telephone +44 (0) 1252 376666.
Fax: +44 (0) 1252 544981

**Solartron Instruments**
11261 Richmond Avenue, Suite #G-105
Houston, Texas, 77082-2615 USA
Telephone: +1 (713) 558 4930
Fax: +1 (713) 558 4701

**Solartron Instruments**
37 Rue du Saule Trapu
91882 Massy, Cedex, France
Telephone: +33 (1) 69 53 63 53
Fax: +33 (1) 60 13 37 06

**Solartron Overseas Sales Ltd**
**Instruments Division**
Block 5012 TECHplace II
Ang Mo Kio Ave. 5, #04-11
Ang Mo Kio Industrial Park
Singapore 2056,
Republic of Singapore
Telephone: +65 482 3500
Fax: +65 482 4645

**Solartron Instruments**
Room 7006, Yuan Shan Hotel
No. 20 Yu Min Tong Li
West District, Beijing 100029
China
Telephone: +86 1-202 9133
Fax: +86 1-202 9133

For details of our agents in other countries, please contact our Farnborough, UK, office.

Solartron Instruments pursue a policy of continuous development and product improvement.
The specification in this document may therefore be changed without notice.

## 7081 WITHIN A SYSTEM

The need for a precision measurement module in a data acquisition, or other processor-controlled, system is readily satisfied by the 7081 voltmeter. All front-panel-selectable facilities (with the exception of the Input Select switch) are fully programmable. The instrument can thus be remotely controlled by electrical commands. An interface – the user has a choice of two international standards – accepts command data from, generates the control signals required by, and outputs data to, the external system. The output information can be simple measurements, processed measurement data, or data accessed from the 7081 history file.

In a simple system, involving the use of just one other device, e.g. a keyboard terminal, the measurement processing and data storage/manipulation facilities of the voltmeter offer considerable sophistication and obviate the need for the terminal to be intelligent. On the other hand, by choosing 7081 for inclusion in a complex, multi-device system such as a fully automated test facility, the instrument's ability to manipulate data can be exploited to complement the computational power of the System Controller. The latter will thus be freed for other tasks, such as servicing the needs of other devices within the system.

The 7081 contains two interfaces either of which can be used on its own, or a system can be configured which combines the facilities that they offer:

1. A two-wire Serial Interface, conforming to the RS232/CCITT V24 standard.

2. A bit-parallel, byte-serial Interface, conforming to IEEE 488(1978) – the GP-IB Interface.

Chapter 1 of this part contains an introduction to interfacing whilst Chapters 2 and 4 provide a brief outline of the two standards and how 7081 utilises them within a system. Chapter 3 is devoted to an explanation of the command languague and error messages.

Digitized by the Internet Archive
in 2025

https://archive.org/details/manualsplus_09561

# Contents Part 2


| | | |
|---|---|---|
| **Chapter 1** | **General Information** | |
| | | *Page No.* |
| 1 | Introduction | 1.2 |
| 2 | Combined System | 1.2 |
| 3 | RS232C(V24) | 1.2 |
| 4 | IEEE 488/1978 (GP-IB) | 1.3 |
| 5 | Command Language | 1.6 |
| **Chapter 2** | **Operation over the GP-IB** | |
| 1 | Introduction | 2.2 |
| 2 | Commanding Remote and Local | 2.2 |
| 3 | Commanding Measurements | 2.2 |
| 4 | Message Protocols | 2.6 |
| 5 | Invalid Characters | 2.6 |
| 6 | Input and Output | 2.7 |
| 7 | GP-IB Functions | 2.7 |
| 8 | Lock Front Panel Command | 2.9 |
| 9 | Interrogation Characters | 2.9 |
| 10 | Processed Measurement | 2.10 |
| 11 | Programming Examples | 2.10 |
| **Chapter 3** | **Command Language** | |
| 1 | Introduction | 3.2 |
| 2 | Commands | 3.2 |
| 3 | Command Execution Order | 3.49 |
| 4 | Error Messages | 3.50 |
| **Chapter 4** | **RS232 Operation** | |
| 1 | Introduction | 4.2 |
| 2 | Hardware and Terminology | 4.2 |
| 3 | Transmission Speeds | 4.2 |
| 4 | Command Language | 4.3 |
| 5 | Input and Output | 4.3 |
| 6 | Programming Examples | 4.4 |
| **Appendix 1** | Examples of Output Formats | A1.1 |
| **Appendix 2** | Adopted Settings | A2.1 |
| **Index** | | |

# Chapter 1
# General Information

**Contents**


| | | *Page No.* |
|---|---|---|
| 1 | Introduction | 1.2 |
| 2 | Combined System | 1.2 |
| 3 | RS232C (V24) | 1.2 |
| 4 | IEEE 488/1978 (GP-IB) | 1.3 |
| 5 | Command Language | 1.6 |

## 1 Introduction

The 7081 interfaces function as bi-directional data transfer devices. Information is received, at an interface, as coded electrical commands and the voltmeter responds by sending either measurements or other data as commanded by the system controller.

Two interfacing standards are available:

1. A serial interface conforming to RS232C.
2. A bit-parallel, byte-serial interface conforming to IEEE 488/1978.

Selection is made simply by connecting 7081 to the system via the appropriate RS232C or IEEE socket at the rear of the unit. The RS232C baud rate is set up by internal switches and the IEEE device address by rear panel switches.

## 2 Combined System

The 7081 is capable of operation within a combined system using both RS232C and IEEE compatible equipment e.g. the voltmeter might be configured within a full IEEE system and yet also be connected, via a separate rear panel connector, to an RS232C terminal.

In a system containing more than one controller, only one can be controller-in-charge at any one time. The other(s) must remain in the idle state until control is handed over. Initially, the controller-in-charge is the first controller to send instructions on the bus.

## 3 RS232C (V24)

RS232C defines a standard interface between data terminal equipment (DTE) and data communication equipment (DCE) employing serial binary data interchange. In this instance, the 7081 acts as a DCE; the DTE might typically be a teleprinter, such as the Texas ASR 743.

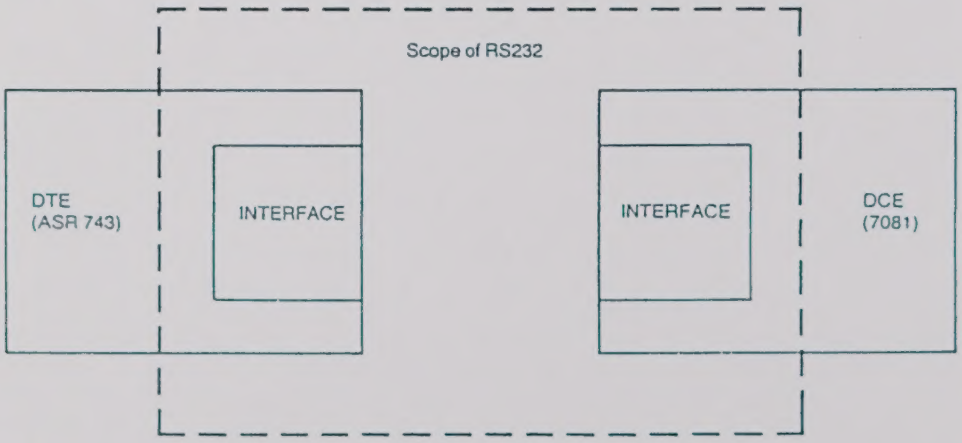



Fig 1.1 *RS232C Linking*

### 3.1 Interface Details

The interface standard, definable by its mechanical and electrical characteristics, specifies fourteen interchange circuit configurations for data transmission. The 7081 RS232C interface is classified as Type E and has the pin/signal assignment listed in Table 1.1.

**Table 1.1** *Pin/Signal Assignment.*

| Pin No | Signal |
|---|---|
| 1 | Protective Ground |
| 2 | Transmitted Data |
| 3 | Received Data |
| 5 | Clear to Send |
| 6 | Data Set Ready |
| 7 | Signal Ground |
| 8 | Received Line Signal Detector |
| 20 | Data Terminal Ready |

All the RS232C control signals (pins 5, 6, 7, 8 and 20) are internally commoned within 7081, thus the voltmeter transmits and receives data over what is effectively a twin-wire system.

## 4 IEEE 488/1978 (GP-IB)

The IEEE 488 interfacing standard, or GP-IB as it is also known, defines a bi-directional bus structure for the interconnection of programmable instruments in byte-serial, bit-parallel interfacing systems. The cables, connectors and control protocols used are rigidly defined to ensure unambiguous interdevice communication. The limitations on a system are:

1. A maximum of 15 devices can be interconnected by a single bus.
2. The total bus length should not be greater than 20m or number of devices ×2m, whichever is the shorter.
3. Transmission rate should not exceed 1 Megabaud.
4. All bus data should be digital.

### 4.1 Devices

Of the 15 devices on the GP-IB, only one is designated controller-in-charge. This device exercises overall bus control and is capable of both receiving and sending data. The remaining devices can be designated as follows:

Listener – able to receive messages
Talker – able to send messages
Talker/Listener Combined – e.g. 7081

The controller can address other devices and command them to listen, address one device to talk, and wait whilst a message is sent. Message routes are set up by the controller but it need not take part in the data interchange.

### 4.2 Cabling

The GP-IB is a standard cable linking the individual instruments into a coherent system. It contains 16 signal wires and 8 earth returns. The cable connectors consist of a plug/socket piggy-back arrangement, so that additional cables can be connected maintaining the parallel bus structure.

The 16 signal wires form the communication highway and comprise:

8 data wires – digital input/output (DIO)
5 management wires – interface control signals
3 handshake wires – transfer of data control

## 4.3 Management

The management lines are used for bus supervision. The signal ATN (Attention) is asserted by the controller whenever it places an address or command message byte on the bus. ATN is used in conjunction with EOI (end or identify) when the controller is carrying out a device poll. The EOI line can also be driven by a talker to indicate the end of a message.

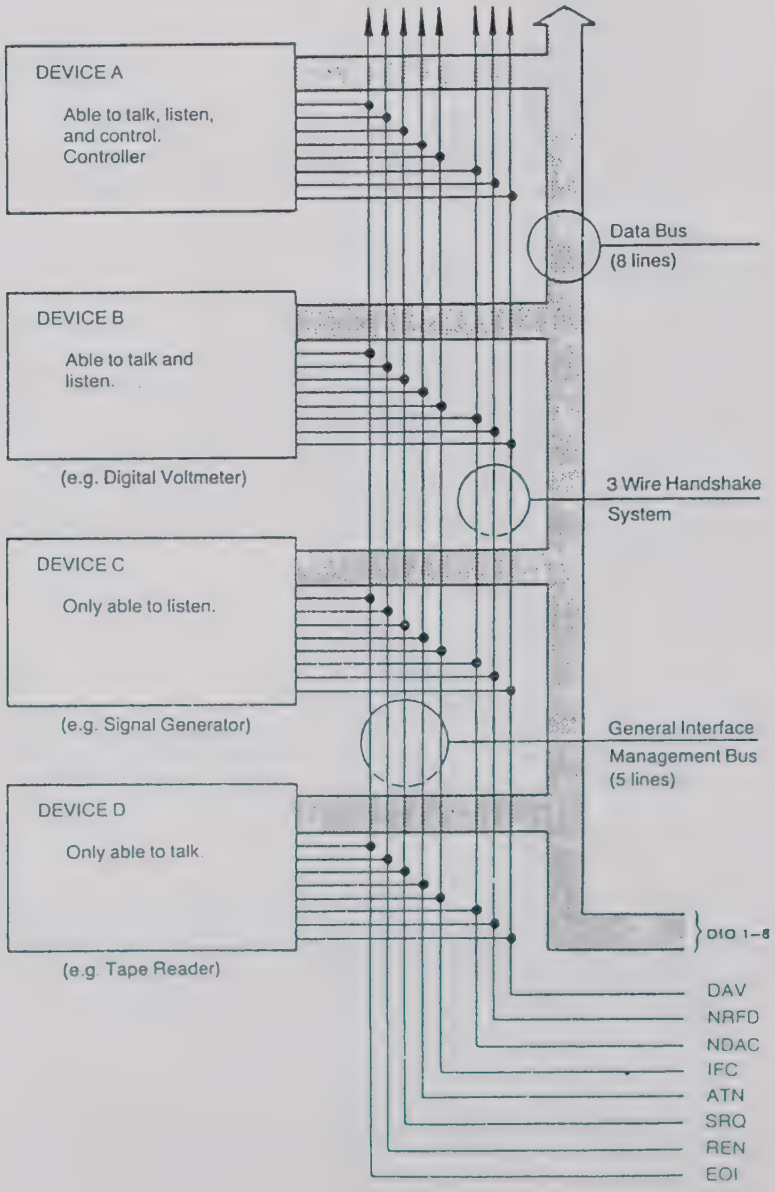



Fig 1.2 *GP-IB Structure*

To alert the controller to the fact that service is required, e.g. data for transfer, a device asserts SRQ. The controller then conducts a device poll to determine which device requires service.

The remaining two lines are largely self-explanatory. IFC (Interface Clear) is used by the controller to initialise the interface to a predetermined state. REN (Remote Enable) is used by the controller to enable an addressed device/devices to accept remote control.

## 4.4 Handshake

Message data is transferred under the control of a 3-wire handshake, which can be arranged to run at the speed of the slowest bus device concerned with the interchange. The talker indicates that a byte is present by asserting DAV (Data Valid), then waits for all devices to indicate that the data has been accepted. The listeners do this by releasing the NDAC (Not Data Accepted) line. Data may only be sent if all devices are ready to receive. This is indicated by the releasing of the NRFD (Not Ready for Data) line by every device as it becomes ready for another data byte.

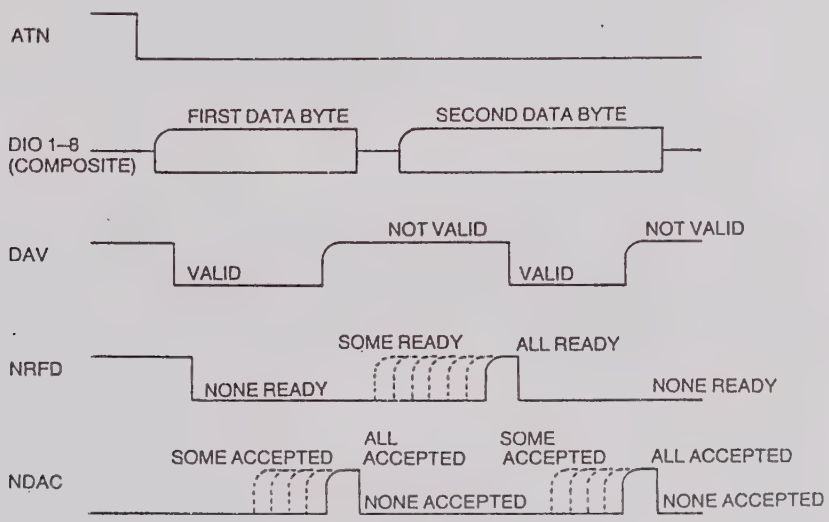



**Fig 1.3** *Handshake Routine Timing*

### 4.5 Address and Talk/Listen Selection

For normal operation on the GP-IB (using a controller) set the selector switch on the rear panel as indicated in Figure 1.4.

ADDRESS: As required

LISTEN ONLY: OFF
TALK ONLY: OFF

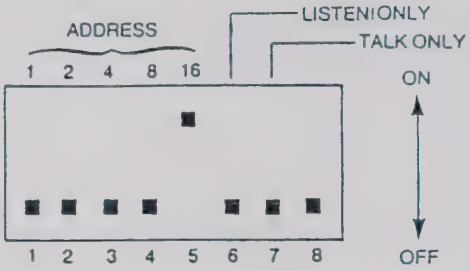



**Fig 1.4** *GP-IB Selector Switches*

For operation without a controller, e.g. if a printer only is connected to 7081, the TALK ONLY mode should be selected. To achieve this, set TALK ONLY to ON.

Note: The voltmeter only looks at the selector switch settings at power-up or initialise.

## 5 Command Language

The command language for both interfaces is formed of English language words, decimal numbers and a set of punctuation symbols. It incorporates, where applicable, the recommended practices set out in the 1981 draft "CODE AND FORMAT CONVENTIONS FOR USE WITH IEEE STANDARD 488 (1978)". The form and type of words (see Chapter 3) are based on those specified in the IEEE standard 416 (1978) commonly known as ATLAS.

# Chapter 2
# Operation over the GP-IB

**Contents**


| | | *Page No.* |
|---|---|---|
| 1 | Introduction | 2.2 |
| 2 | Commanding Remote and Local | 2.2 |
| 3 | Commanding Measurements | 2.2 |
| 4 | Message Protocols | 2.6 |
| 5 | Invalid Characters | 2.6 |
| 6 | Input and Output | 2.7 |
| 7 | GP-IB Functions | 2.7 |
| 8 | Lock Front Panel Command | 2.9 |
| 9 | Interrogation Characters | 2.9 |
| 10 | Processed Measurement | 2.10 |
| 11 | Programming Examples | 2.10 |

This chapter describes the use of 7081 within a GP-IB system.

## 1 Introduction

All 7081 facilities can be accessed and controlled remotely via the GP-IB using a suitable controller. These facilities consist of all those offered at the front panel plus some additional ones.

The 7081 will accept commands from the controller, initiate measurements and send results back to the controller when requested. In addition the 7081 can be programmed to interrupt the controller when a measurement or an error occurs, thereby allowing the controller to conduct other tasks until interrupted and so increase system efficiency.

Connection to the GP-IB is made using the IEEE 488/GP-IB Interface socket on the rear panel. The 7081 address and Talk/Listen status are selected using the adjacent switch.

## 2 Commanding Remote and Local

Depending upon the interface switch settings and certain GP-IB commanded parameters, 7081 can be switched between local and remote operation either manually, or by electrical commands. With full IEEE 488/1978 implementation, i.e. GP-IB ON and 7081 not in TALK ONLY:

1. On power-up the instrument is in local operation without Local Lockout.

2. With REN (Remote Enable) asserted and after the listen address has been received from the GP-IB controller, 7081 will be in remote.

3. Local Lockout disables the local control preventing any attempt to return to local via the front panel. The RS232 interface is also disabled by this command.

4. If the instrument, in the remote state, receives the GP-IB command GO TO LOCAL, or REN is unasserted, 7081 enters the local state.

The remainder of this description covers the operation of 7081 in remote.

## 3 Commanding Measurements

With 7081 under remote control, front panel selection of function, range, scale length etc. is not possible. Therefore, the controller must command all the settings and instruct the voltmeter to take a measurement. Instructions are sent in the form of a message string, using the appropriate commands followed by a number or word which clearly defines the setting. For example, to command 7081 to operate on the 100 range, the command is RANge=100; if Autorange is required the command is RANge=Auto.

Other parameters are commanded in a similar fashion; measurement function is, for example MODe=VDC; displayed scale length is NInes=n where n=3 to 8 Thus.

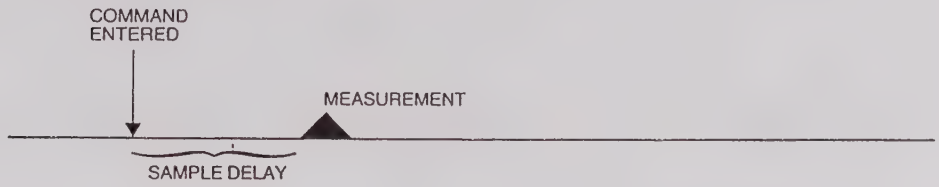
MODe=VDC: RANge=Auto: NInes=6

commands the voltmeter to select Vdc, Autorange, 6×9's displayed.

The Measure command is used to instruct the voltmeter to take a measurement. The command can take several forms:

1. **MEAsure, SIngle**
   **or**
   **MEAsure, 1**

   Both these commands are equivalent and cause a measurement to be made on the present settings. The measurement is preceded by a sample delay appropriate to those settings or a user defined delay, if required.

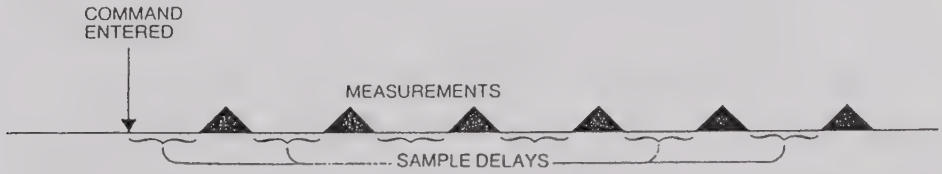



2. **MEAsure, COntinuous**

   This command causes successive measurements to be taken at the fastest possible rate for the integration time selected. There is a sample delay before measurements begin, but, once measurements have begun, there is no inter measurement delay. Sample delays are only implemented if there is a measurement setting change (mode or range).

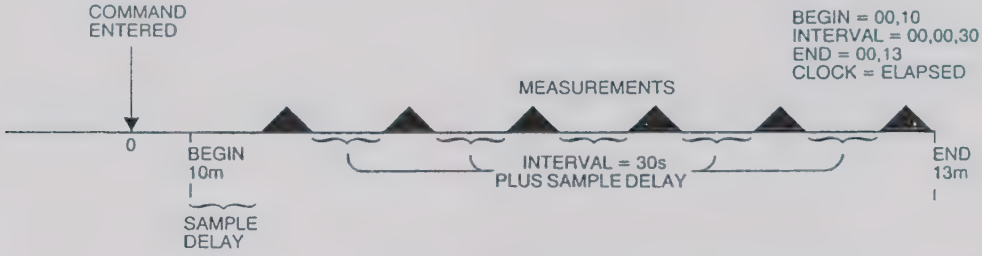
3. **MEAsure, 6**

   Six measurements are produced on the present settings. There is a sample delay between each measurement.



**4. MEAsure, CLock controlled**

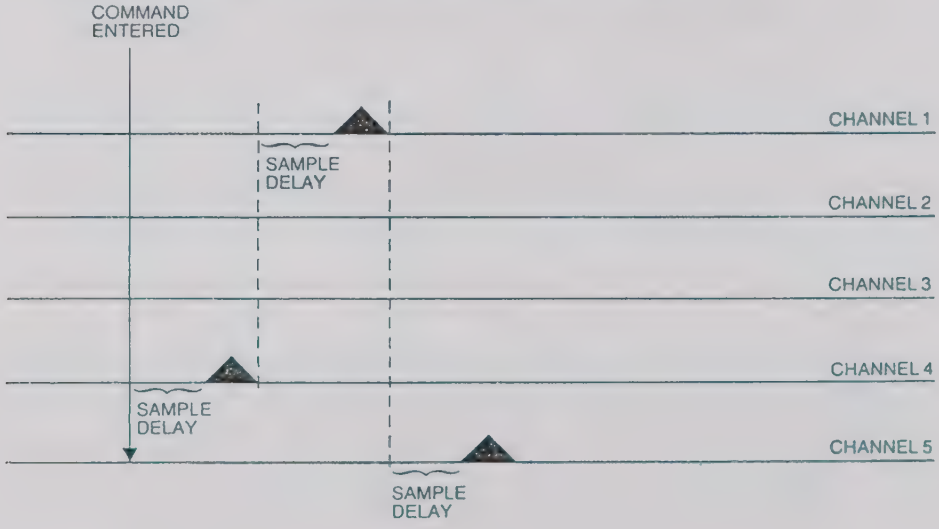
Single measurements are generated at defined times, the total number of measurements taken depending on the time values set up via the Begin, Interval, End and Clock commands




When operating 7081 with a multi-channel scanner, such as Solartron Minate (7010), channel measurements can be taken using the commands given below.

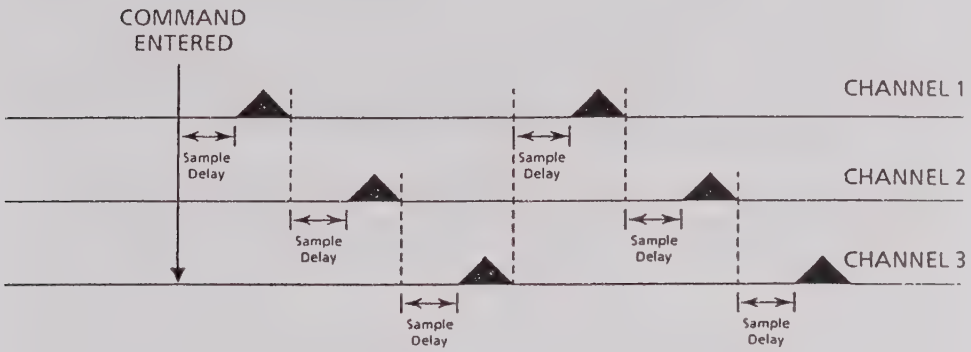
**5. MEAsure, CHannel, 4, 1, 5**

A single measurement is performed on each of the channels specified. Channel pull-in and drop-out delays may be specified under the Channel command and are in addition to the sample delays. The voltmeter is left on the last channel selected.



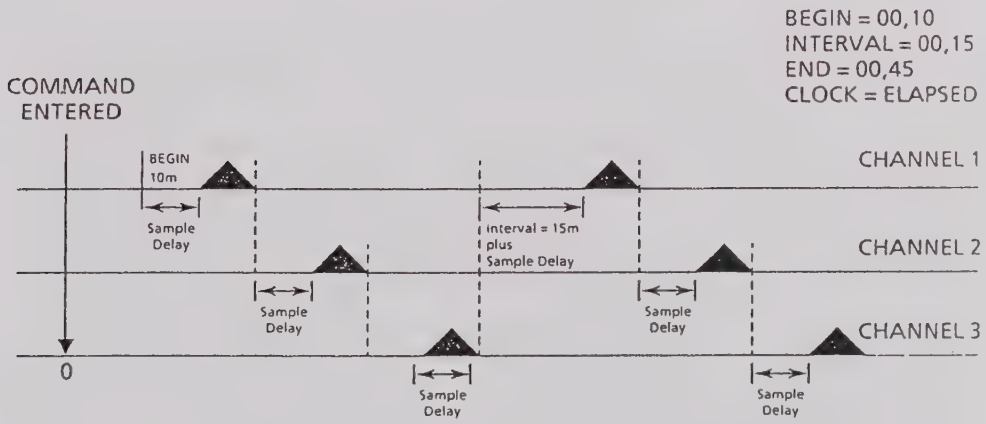
6. **MEAsure, 2, CHannel, 1, To, 3**

A single measurement is made of each specified channel and then the process is repeated until the specified number of channel scans, i.e. 2, has taken place.




7. **MEAsure, CLock controlled, CHannel, 1, To, 3**

In this command each event commanded by the clock control is a channel scan.




8. **TRigger**

This command initiates a measurement event which has been 'armed', i.e.

MEAsure, 9, ARM

ARM may be used with facilities 3, 4, 5, 6 and 7, and has the meaning "waiting for trigger". If no facility has been armed, TRigger assumes the meaning MEAsure SIngle.

It is not always possible to determine how much output will be generated as a result of a Measure command, e.g.

If an interval time for clock controlled measurements is shorter that the event takes to be performed, fewer readings than expected will be obtained. In this case, the following command could be used:

MEAsure, CLock : MEAsure?

As Measure commands are actioned sequentially, the reply to MEAsure? only appears after the clock control has finished.

MEAsure, COntinuous can be stopped using the MEAsure, STop command, however, although no more measurements are produced, there may be results 'in the pipeline'. If no more output is required, STOp should be used as this command clears both the input and output queues.

## 4 Message Protocols

For 7081 to accept the controller's message and act upon its instructions, certain conditions must be met:

1. The voltmeter must receive the message.

2. The message must be recognisable as being a command.

3. A terminator must follow the message.

Condition 1 is satisfied by the GP-IB controller asserting REN and sending MLA (My Listen Address), i.e. addressing 7081 as a Listener.

Condition 2 in the GP-IB system calls for the unassertion of ATN (Attention), the signal which identifies the information as a Device Dependent Message.

Condition 3. Command messages, of any type and irrespective of their origin, are received by the interface and stored in a queue. Up to 76 characters can be accepted as a single command string. The 7081 will only action the command string on receipt of:

LF Line Feed character
or
EOI with any character

## 5 Invalid Characters

As a general rule, 7081 will ignore commands etc. not included in the command language. If required however, the interface can be programmed to output an error message over the GP-IB or assert SRQ if it receives an invalid command or an invalid command argument.

## 6 Input and Output

If the controller attempts to input messages to the 7081 at a faster rate than the unit can implement them, the 7081 will slow down the handshake routine thereby slowing the controller's ability to input messages.

The output of 7081 will queue, in general, up to three messages before ceasing operation, i.e. measurements, processing, etc., if the connected devices do not take the output. Once the queued output has been taken, 7081 will continue where it left off. The voltmeter will still accept up to three input messages, if the output is queued.

**Note:** A message is defined, in this context, as an input line terminated by a Carriage Return, Line Feed or EOI.

The following commands are relevant to output control:

1. OUTPUT – allows generated data to be routed only to the interfaces desired. If both RS232 and GP-IB are on, output occurs at the rate of the slowest device to ensure that both outputs remain synchronised.

2. FORMAT – controls the format of the measurement results output.

3. CAPITALS LOCK – particularly useful with primitive controllers.

4. DELIMIT – defines the 'end of line' character.

5. ERROR – if the GP-IB output is on and ERror = Verbose, error reporting messages will also appear over the bus.

6. SRQ – enables a Service Request to be generated on Ready, Output Available, Error, etc.

These commands, together with specific program commands, enable powerful editing of information transmitted to the output and into the history file.

## 7 GP-IB Functions

Two messages, Service Request and Parallel Poll Configure, are actioned only within the GP-IB. They enable: 7081 to request service from the GP-IB controller by means of the SRQ line; the unit's DIO lines to be configured, i.e. coded, for Parallel Poll interrogation by the controller.

### 7.1 Parallel Poll

In a system containing more than one controlled device it is essential that the controller should be able to identify which instrument is requesting service when SRQ is asserted. It obtains this information by conducting a Parallel Poll.

The systems interface is configured for parallel poll in accordance with sub-set PP1 of the IEEE 488/1978 standard.

The 7081 interface allows for the voltmeter to be allocated a unique data wire (DIO 1 through 8) as its parallel poll code wire.

In the multi-device system under consideration, if an interrupt occurs as a result of SRQ, the controller must initiate a parallel poll using the GP-IB IDENTIFY (IDY) command.

In configuring for parallel poll, the controller allocates each instrument one of the eight DIO wires as its code wire. When the IDY command is obeyed, the data sent back to the controller is a representation of the 8 lines indicating which devices are responding, i.e.

0 0 0 0 0 0 0 1

indicates that the device allocated DIO 1 is requesting attention.

To complete its response to the SRQ interrupt, the controller must interrogate (Serial Poll) the device requesting service, by sending SPE as described below.

## 7.2 Parallel Poll Sharing

Where the number of controlled devices exceeds eight (the number of DIO lines available), it is not possible to allocate each device an individual DIO wire for polling purposes. Thus, if may be necessary for two devices to share a common line. A parallel poll will establish which DIO line has been set, then serial polling will ascertain which of the two devices has requested service, and what service is required.

## 7.3 Serial Poll

When the controller conducts a Serial Poll it sends SPE (Serial Poll Enable) along with a talk address to 7081 which responds by outputting the value of its serial poll byte.

It is recommended that, for maximum value in a computer-controlled system, the Serial Poll instruction is used as part of a user-program subroutine.

## 7.4 Serial Poll Byte

Voltmeter status is coded in an 8-bit register in the interface as the Serial Poll Byte, which is made up as follows:

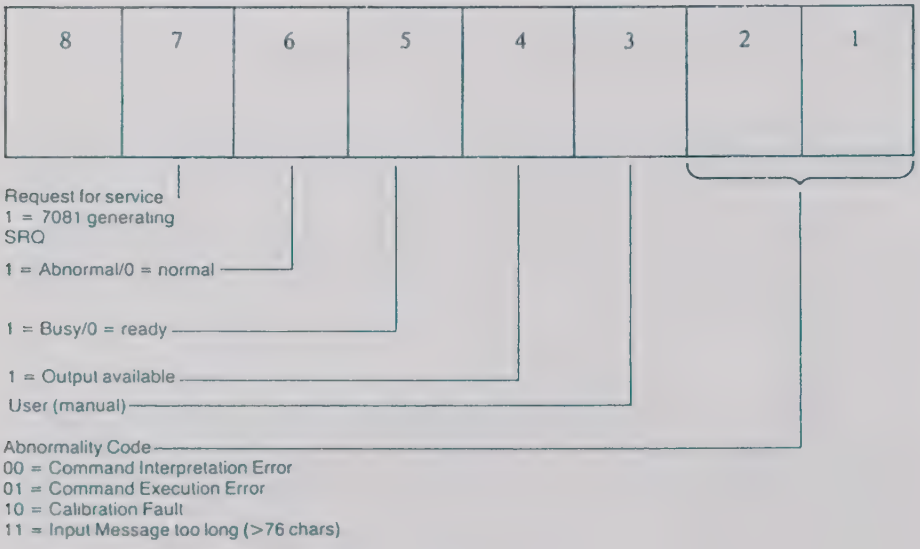



**Fig 2.1** *Serial Poll Byte*

The 7081 requests service by setting the rqs bit true and asserting the SRQ line. The user's software must include the means of detecting, and reacting to SRQ. The SRQ (and rqs) facility is controlled by the SRQ command, where SRq=OFF inhibits SRQ; Error, Output, User and Ready define what will cause an SRQ.

Output is an 'enable' for SRQ when data, control information, error messages are available at the GP-IB interface; bit 4 of the serial poll byte is set, 7081 asserts SRQ and sets bit 7 true.

User enables the front panel SRQ control; bit 3 of the serial poll byte is set, 7081 asserts SRQ and sets bit 7 true.

Ready enables SRQ when the unit is not busy; bit 5 of the serial poll byte is set to zero, 7081 asserts SRQ and sets bit 7 true. SRQ will only be asserted when all of the instrument is ready.

### 7.5 Error Indication

The value of the serial poll byte will be modified if SRQ is generated as a result of the interface receiving an invalid message. Although the voltmeter itself ignores the setting commanded by the invalid message, the error is detected by the interface and SRQ is asserted. At the same time, the error type is coded in bits 1 and 2 of the serial poll byte and rqs is set true. In the 7081 interface, the possible codes are:

00 = Command Interpretation Error
01 = Command Execution Error
10 = Calibration Fault
11 = Input Message too long (> 76 chars.)

## 8 Lock Front Panel Command

When 7081 is functioning under remote control, it is possible, under certain conditions, for an operator to resume local control by pressing the front panel Local key. For controllers which do not implement Local Lockout, this can be prevented by inhibiting the keyboard with the command: LOck front panel=ON. The front panel controls are enabled by: LOck front panel=OFF.

## 9 Interrogation Characters

It is desirable to have a means whereby the commanded settings can be verified. 7081's settings can be accessed individually, by means of the query character, e.g.:

MODe? – interrogates the measurement function
RANge? – interrogates the selected range

to which the unit will respond with a status reply.

## 10 Processed Measurement

The processing commands are sent as command strings in exactly the same way as those commanding unprocessed measurement. Thus, for program definition, the message:

RATio, MOde=Main/N, N=5.6, ON

can be sent. This defines the Ratio program to mode main/N with a value of 5.6 for N. The command also turns the program on.

### 10.1 History Dump

If the user wishes to access information stored within the instrument, the Dump command should be used. This command initiates output of the unit history file contents.

The Dump command is also useful in determining how many results a program has produced. For example, if the Limits program is configured to output only no go results, the number of results out of range will be indeterminate. The readings could be stored in the history file and upon completion, the command DUmp? could be used to indicate the number of readings to dump.

### 10.2 Program Exit

The 7081 can be commanded to perform a mixture of unprocessed and processed measurements. Exit from processed measurement is commanded by COmpute, OFF, i.e. the instrument reverts to unprocessed measurement. COmpute, ON causes 7081 to resume processed measurement.

## 11 Programming Examples – HP9835A

In the following examples of controlling the 7081 via the GP-IB, the system controller is assumed to be a HP9835A calculator. All command messages are, therefore, in the operating code and format of that device.

**Example 1** Single measurement under remote control

```
10 DIM B$[80]
```
*This instruction is internal to the controller. It defines a string variable, B$, of length 80 characters.*

```
20 RESET 716
```
*Clears the interface to a pre-defined state and 7081 to its power-up (resumed) conditions.*

```
30 OUTPUT 716; "INITIALISE"
```
*Initialises the 7081.*

```
35 WAIT 3000
```
*Wait for 7081 to complete initialisation. (Value in milliseconds.)*

```
40 OUTPUT 716; "OUTPUT, GP-IB, ON: FORMAT = ENGINEERING,EXPANDED"
```
*This command line turns on the GP-IB output and defines the format of the output.*

```
50 OUTPUT 716; "MODE=VAC: RANGE=100: NINES=7"
```
*Defines the mode, range and scale length for 7081 to use.*

```
60 OUTPUT 716; "MEASURE, SINGLE"
```
*Instructs 7081 to perform a single measurement*

```
70 ENTER 716; B$
```
*These two commands store the result in B$ within the controller and then access the store and print the result.*

```
80 PRINT B$
```

```
90 STOP
```
*Stop is a local message to the controller, terminating the program.*

To execute the program, press the controller RUN button.

**Example 2** Using the Ratio program

```
10 DIM N$ [90]
```
*Internal string definition.*

```
20 RESET 716
```
*As in Example 1.*

```
30 OUTPUT 716; "INITIALISE"

35 WAIT 3000

40 OUTPUT 716; "OUTPUT, GP-IB, ON: FORMAT = DVM, COMPRESSED"
```
*This command line turns on the GP-IB output and defines the format of the output.*

```
50 OUTPUT 716; "RATIO, MODE = MAIN/N, N=6, ON"
```
*Defines the Ratio program for 7081.*

```
60 OUTPUT 716; "MEASURE, 15"
```
*Instructs 7081 to take 15 measurements and process them.*

```
70 FOR I = 1 TO 15
```
*Enters into the controller and prints each of the 15 results.*

```
80 ENTER 716; N$

90 PRINT N$

100 NEXT I

110 STOP
```
*Program terminator.*

Fifteen results are output, when this program is run, indicating the ratio of the main input terminals to the value N=6.

**Example 3** This example uses the Statistics program to obtain the average of a set of results, and then scales these averages.

```
10 DIM C$ [70]

20 RESET 716

30 OUTPUT 716; "INITIALISE"

35 WAIT 3000

40 OUTPUT 716; "OUTPUT, GP-IB, ON: FORMAT=ENGINEERING, EXPANDED"

50 OUTPUT 716; "STATISTICS, MODE = WINDOW, WINDOW SIZE=10,
   OUTPUT=AVERAGE, ON"        Defines the Statistics program for 7081.

60 FOR I = 1 TO 10            This command sequence produces 10 averages,
                              multiplies each one by 3 and adds 2. The average is
                              then displayed at the output followed by the result of
                              the scale equation.

70 OUTPUT 716; "MEASURE, 10"

80 ENTER 716; C$

90 PRINT C$

100 LET X = VAL(C$)

110 LET Y = 3*X+2

120 PRINT Y

130 NEXT I

140 PRINT "COMPLETE"

150 STOP
```

**Example 4** Using the Scale and Limits programs

```
10 DIM A$[70],B$[70]

20 RESET 716

30 OUTPUT 716; "INITIALISE"

35 WAIT 3000

40 OUTPUT 716; "OUTPUT,GP-IB, ON: FORMAT=DVM, EXPANDED"

50 OUTPUT 716; "MODE=VAC"      Sets the 7081 measurement function to volts ac.

60 OUTPUT 716; "SCALE,M=2, C=4,ON"
                               Defines the Scale program.

70 OUTPUT 716; "MEASURE,4"     Initiates and prints 4 results

80 FOR I = 1 TO 4

90 ENTER 716; A$

100 PRINT A$

110 NEXT I

120 OUTPUT 716; "LIMITS, MODE=WINDOW, HI LIMIT = 10, LO LIMIT = 2,
    WINDOW SIZE = 4, OUTPUT = NUMBER GO, ON"
                               Defines the Limits program

130 FOR J=1 TO 4               Initiates and prints 4 results, which are passed
                               through both the Scale and Limits programs

140 OUTPUT 716; "MEASURE,4"

150 ENTER 716; B$

160 PRINT B$

170 NEXT J

180 PRINT "COMPLETE"

190 STOP
```

## 11.1 Clock Controlled Measurements

The following four example programs show the different techniques that can be used when initiating clock controlled measurements. The basic programming techniques could also be applied to other commands.

**Example 1** This program sets up the clock and takes measurements which are then output and displayed at the controller. To implement the program the user must know in advance how many results will be produced as this number is used for the FOR . . . NEXT loop.

```
10 DIM B$[80]                    Defines a string variable, B$, of length 80 characters.
                                 This command is internal to the controller.

20 RESET 716                     Clears the interface to a pre-defined state and 7081 to
                                 its power-up (resumed) conditions.

30 OUTPUT 716; "INITIALISE"      Initialises the 7081.

40 WAIT 3000                     Defines a waiting time to enable the 7081 to complete
                                 initialisation.

50 REMOTE 716                    Puts the 7081 into the remote state.

60 LOCAL LOCKOUT 7               Disables the 7081 'local' control.

70 OUTPUT 716; "OUTPUT,GP-IB, ON: FORMAT = EXPANDED"
                                 Turns on the voltmeter GP-IB output and sets the
                                 result format to expanded.

80 OUTPUT 716; "NINES = 5"       Defines the 7081 scale length and integration time. At
                                 power-up the voltmeter adopts mode = Vdc, range =
                                 Auto so these commands do not have to be specified.

90 OUTPUT 716; "BEGIN = 0,0,30"  Sets up the start time for the clock to 30 seconds after
                                 clock control activation.

91 OUTPUT 716; "INTERVAL = 0,0,30"
                                 Defines the interval between measurements as 30
                                 seconds.

92 OUTPUT 716; "END = 0,5,30:CLOCK=ELAPSED"
                                 Sets up the end time as five minutes, 30 seconds after
                                 the start and defines the clock time as elapsed time as
                                 opposed to real time.

100 OUTPUT 716; "MEASURE, CLOCK CONTROLLED"
                                 Activates measurements under the control of the clock
                                 set up in lines 90, 91 and 92.

110 FOR I=1 TO 11                This FOR . . . NEXT loop reads a measurement from
                                 the voltmeter into the string variable defined in line
120 ENTER 716;B$                 10, prints out the result and then passes on to the next
                                 measurement.

130 PRINT B$

140 NEXT I

150 STOP                         Terminates the program
```

**Example 2** In the event that the number of results a program will produce is not known, the following technique could be used. The query MEASURE? is used to inform the controller when the 7081 has ceased measuring.

| | |
|---|---|
| 10 DIM B$ [80] | *Defines a string variable, B$, of length 80 characters. This command is internal to the controller.* |
| 20 RESET 716 | *Clears the interface to a pre-defined state and 7081 to its power-up (resumed) conditions.* |
| 30 OUTPUT 716; "INITIALISE" | *Initialises the 7081.* |
| 40 WAIT 3000 | *Defines a waiting time to enable the 7081 to complete initialisation.* |
| 50 REMOTE 716 | *Puts the 7081 into the remote state.* |
| 60 LOCAL LOCKOUT 7 | *Disables the 7081 'local' control.* |
| 70 OUTPUT 716; "OUTPUT, GP-IB, ON: FORMAT = EXPANDED" | *Turns on the voltmeter GP-IB output and sets the result format to expanded.* |
| 80 OUTPUT 716; "NINES = 5" | *Defines the 7081 scale length and integration time. At power-up the voltmeter adopts mode = Vdc, range = Auto so these commands do not have to be specified.* |
| 90 OUTPUT 716; "BEGIN = 0,0,30" | *Sets up the start time for the clock to 30 seconds after clock control activation.* |
| 100 OUTPUT 716; "INTERVAL = 0,0,30" | *Defines the interval between measurements as 30 seconds.* |
| 110 OUTPUT 716; "END = 0,5,30: CLOCK = ELAPSED" | *Sets up the end time as five minutes, 30 seconds after the start and defines the clock time as elapsed time as opposed to real time.* |
| 120 OUTPUT 716; "MEASURE, CLOCK CONTROLLED " | *Activates measurements under the control of the clock set up in lines 90, 100 and 110.* |
| 130 OUTPUT 716; "MEASURE?" | *Queries the state of the measurements, i.e., when the end time has been reached the unit will cease measuring and MEASURE? will produce the reply MEASURE = STOP.* |
| 140 ENTER 716; B$ | *Enters each measurement into B$* |
| 150 IF UPC$ (B$) = "MEASURE = STOP" THEN 180 | *When 7081 outputs MEASURE = STOP into B$ the program will jump to the terminator.* |
| 160 PRINT B$ | *Prints out each measurement placed in B$ except MEASURE = STOP.* |
| 170 GO TO 140 | *Goes onto the next measurement.* |
| 180 STOP | *Program terminator.* |

**Example 3** In this program, 7081 uses its Serial Poll Byte to signal to the controller when it has data for output. The controller performs a background routine and 7081 interrupts this routine when it has completed its measurements. To use a program of this type the controller must be able to conduct both a parallel and a serial poll.

```
10 DIM A$ [100]
```
*Defines a string variable, A$, of length 100 characters.*

```
20 S=0! SERIAL POLL HOLDING REG
```
*Defines the serial poll register and sets it to zero.*

```
30 P=0! PARALLEL POLL HOLDING REG
```
*Defines the parallel poll register and sets it to zero.*

```
40 Waiting = 0! BACKGROUND WAITING FOR SIGNAL FLAG
```
*Sets the Waiting flag, for the background routine, to zero.*

```
50 Signal = 0! INTERRUPT ACKNOWLEDGE FLAG
```
*Sets the Interrupt Acknowledge flag, for the background routine, to zero.*

```
60 Rqs = 6! SERPOL BIT — REQUEST FOR SERVICE
```
*Defines the Serial Poll Request for Service bit.*

```
70 Out = 3! SERPOL BIT — OUTPUT AVAIABLE
```
*Defines the Serial Poll Output available bit.*

```
80 RESET 716
```
*Clears the interface to a predefined state and 7081 to its power-up (resumed) conditions.*

```
90 OUTPUT 716; "INITIALISE"
```
*Initialises the 7081.*

```
100 WAIT 3000
```
*Defines a waiting time to enable the 7081 to complete initialisation.*

```
110 GOSUB Setremote
```
*Switches the program to the 'Setremote' subroutine.*

```
120 ON INT #7 GOSUB Intserve! DEFINE INTERRUPT SERVICE ROUTINE
```
*When the 7081 interrupts the controller the program switches to the 'Intserve' subroutine.*

```
130 CONTROL MASK 7; 128! UNMASK GP-IB INTERRUPT
```
*Unmasks the GP-IB Interrupt bit.*

```
140 CARD ENABLE 7! ENABLE INTERRUPT
```
*Enables the interrupt card in the controller.*

```
150 OUTPUT 716; "OUTPUT, GP-IB, ON: FORMAT = EXPANDED"
```
*Turns on the voltmeter GP-IB output and sets the result format to expanded.*

```
160 OUTPUT 716; "SRQ, OUTPUT AVAILABLE, ON"
```
*Enables the voltmeter service request bit and defines it to be asserted when the 7081 has output available at the interface*

```
170 OUTPUT 716; "NINES = 5"
```
*Defines the 7081 scale length and integtration time. At power-up the voltmeter adopts mode = Vdc, range = Auto so these commands do not have to be specified.*

```
180 OUTPUT 716; "BEGIN = 0,0,10"
```
*Sets up the start time for the clock to 10 seconds after a trigger.*

```
190 OUTPUT 716; "INTERVAL = 0,0,05"
```
*Defines the interval between measurements as 5 seconds.*

```
200 OUTPUT 716; "END = 0,1,30 : CLOCK=ELAPSED"
```
*Sets the end time as one minute, 30 seconds after the start and defines the clock time as elapsed time as opposed to real time.*

```
210 OUTPUT 716; "MEASURE, CLOCK CONTROLLED"
```
*Activates measurements under the control of the clock set up in lines 180, 190 and 200.*

```
220 OUTPUT 716; "MEASURE?"
```
*Queries the state of the measurements, i.e. when the end time has been reached the unit will cease measuring and MEASURE? will receive the reply MEASURE=STOP.*

```
230 Run:!
```
*Defines a program label to return to after a subroutine.*

```
240 GOSUB Waitsignal
```
*Switches the program to the 'Waitsignal' subroutine.*

```
250 IF UPC$(A$) = "MEASURE=STOP" THEN GOTO Stop
```
*When 7081 outputs MEASURE=STOP into A$ the program will jump to the line labelled Stop.*

```
260 GOTO Run
```
*Returns the program to the line labelled Run.*

```
270 Stop:!
```
*Defines a program label.*

```
280 DISP "END"
```
*Prompts the controller to display 'END'*

```
290 STOP
```
*Program terminator*

```
300!
310!
320!
```
*These lines just supply gaps in the program between the main body of the program and the subroutines.*

```
330 Waitsignal: Waiting = 1 ! SET WAITING FLAG
```
*This line supplies both a label for the 'Waitsignal' subroutine, i.e. marks the beginning, and sets the Waiting flag to 1.*

```
340 Idle = 0
```
*Sets the controller Idle flag to zero.*

```
350 Repeatwait:! REPEAT
```
*Defines a program label to return to.*

```
360 Idle = Idle + 1
```
*Increments the Idle flag.*

370 DISP "STATUS: WAITING FOR INTERRUPT"; Idle

*Prompts the controller to display 'Status: Waiting for Interrupt' followed by the value of the Idle flag.*

380 IF Signal = 0 THEN GOTO Repeatwait! REPEAT UNTIL SIGNAL SET

*Switches the program back to the line labelled 'Repeatwait' unless the Interrupt Acknowledge flag has the value 1.*

390 Signal = 0 — *Resets the Interrupt Acknowledge flag.*

400 PRINT" " — *Prompts the controller to print out a blank line.*

410 RETURN — *Switches the program back to the line after the subroutine was called, i.e. line 250.*

420! INTERRUPT SERVICE ROUTINE

*This line serves no programming function. It simply serves as a title used to describe the subroutine which follows.*

430 Intserve:! — *Label line which marks the beginning of the Interrupt Service routine.*

440 Int = Int + 1 — *Defines the number of the Interrupt.*

450 PRINT "INTERRUPT #"; Int — *Prompts the controller to print out 'Interrupt #' followed by the number of the interrupt.*

460 PPOLL CONFIGURE 716; "00001011"! BIT 3, SENSE TRUE

*Assigns Bit 3 sense true to the 7081.*

470 P = P POLL 7! CONDUCT POLL

*Causes the controller to perform a parallel poll.*

480 GOSUB Remotelocal — *Switches the program to the 'Remotelocal' subroutine.*

490 IF BIT (P,3) <> 1 THEN GOTO Endpol ! 7081 REQUESTING SERVICE?

*Switches the program to the end of the polling sequence if bit 3 is not true during a parallel poll*

500 STATUS 716;S — *The controller conducts a serial poll*

510 IF BIT (S,Rqs) <> 1 THEN GOTO Endpol

*Switches the program to the end of the polling sequence if the Request for Service bit is not set to 1 during a serial poll.*

520 IF BIT (S, Out) <> 1 THEN GOTO Endtry! OUTPUT AVAILABLE?

*Switches the program to the end of the polling sequence if the Output Available bit is not set to 1 during a serial poll.*

530 PRINT "OUTPUT AVAILABLE:";

*Prompts the controller to print out 'Output Available'.*

| | |
|---|---|
| 540 ENTER 716; A$ | *Enters each measurement into A$.* |
| 550 PRINT A$ | *Prints out each measurement placed in A$.* |
| 560 Signal = 1! SIGNAL BACKGROUND | *Sets the Interrupt Acknowledge flag to 1 to indicate that the interrupt has been attended to.* |
| 570 Endtry:! | *Line lable signifying the end of an interrupt.* |
| 580 IF Waiting = 0 THEN GOTO Endwait! IF BACKGROUND WAITING | *Switches the program to the line labelled Endwait if the 'Waiting for Signal' flag is reset.* |
| 590 Waiting = 0! THEN CLEAR WAITING FLAG | *Reset the Waiting flag if set.* |
| 600 Endwait:! | *Line label signifying Waiting flag reset.* |
| 610 Endpol:! | *Line label signifying end of polling sequence.* |
| 620 PPOLL UNCONFIGURE 716 | *Unconfigures the parallel poll previously set up.* |
| 630 GOSUB Remotelocal | *Switches the program to the "Remotelocal" subroutine.* |
| 640 CARD ENABLE 7! RE-ENABLE INTERRUPT | *Re-enables the interrupt card in the controller.* |
| 650 RETURN | *Switches the program back to the line after the subroutine was called, i.e. line 130.* |
| 660 Setremote:! | *Line label marking the beginning of the 'Setremote' subroutine.* |
| 670 Remote = 1! SET REMOTE FLAG | *Sets the 'Remote' flag to 1.* |
| 680 REMOTE 716 | *Puts the 7081 into the remote state.* |
| 690 LOCAL LOCKOUT 7 | *Disables the 7081 'local' control* |
| 700 RETURN | *Switches the program back to the line after the subroutine was called, i.e. line 120 and 780* |
| 710 Setlocal:! | *Line label marking the beginning of the 'Setlocal' subroutine.* |
| 720 Remote = 0! CLEAR REMOTE FLAG | *Resets the 'Remote' flag to zero.* |
| 730 LOCAL 7 | *Puts the 7081 into the local state* |
| 740 RETURN | *Switches the program back to the line after the subroutine was called, i.e. line 800* |

750 Remotelocal:!

*Line label marking the beginning of the 'Remotelocal' subroutine. This routine is needed because the HP9835A returns devices to local during a parallel poll.*

760 IF Remote = 0 THEN GOTO Elseremote! IF STATE WAS REMOTE

*Switches the program to the line labelled 'Elseremote' if the 7081 was previously in local.*

770 GOSUB Setremote! THEN RETURN TO REMOTE

*Switches the program to the 'Setremote' subroutine if the 7081 was previously in remote.*

780 GOTO Endremote

*Switches the program to the line labelled 'Endremote'*

790 Elseremote: GOSUB Setlocal! ELSE RETURN TO LOCAL

*Line labelled 'Elseremote' which switches the program to the 'Setlocal' subroutine.*

800 Endremote:!

*Line label signifying the end of the 'Remotelocal' subroutine.*

810 RETURN

*Switches the program back to the line after the subroutine was called, i.e. line 490.*

**Example 4** In this program, 7081 signals the controller via the Serial Poll Byte, that it has completed the measurements. The measurements are then sent through the Limits program to find the peak to peak value. When 7081 has completed this computation it signals to the controller via its SRQ, READY command. To use a program of this type the controller must be able to conduct both a parallel and a serial poll.

Interrupts for SRQ on Abnormal, Output Available and User are also contained in the program but they are not used.

10 DIM A$ [100]

*Defines a string variable, A$, of length 100 characters.*

20 S=0! SERIAL POLL HOLDING REG

*Defines the serial poll register and sets it to zero*

30 P=0! PARALLEL POLL HOLDING REG

*Defines the parallel poll register and sets it to zero*

40 Waiting = 0! BACKGROUND WAITING FOR SIGNAL FLAG

*Sets the Waiting flag, for the background routine, to zero.*

50 Signal = 0! INTERRUPT ACKNOWLEDGE FLAG

*Sets the Interrupt Acknowledge flag to zero*

60 Rqs = 6! SERPOL BIT — REQUEST FOR SERVICE

*Defines the Serial Poll Request for Service bit.*

70 Abn = 5! SERPOL BIT — ABNORMAL

*Defines the Serial Poll Abnormal bit*

80 Abnmask = 3! SERPOL ABNORMAL CODE MASK

*Defines the Serial Poll Abnormal Code Mask bit.*

90 Rdy = 4! SERPOL BIT — READY (0=READY)

*Defines the Serial Poll Ready bit.*

100 Out = 3! SERPOL BIT — OUTPUT AVAILABLE

*Defines the Serial Poll Output Available bit.*

110 Usr = 2! SERPOL BIT — USER/FRONT PANEL

*Defines the Serial Poll User bit.*

120 RESET 716 — *Clears the interface to a pre-defined state and 7081 to its power-up (resumed) conditions.*

130 OUTPUT 716; "INITIALISE" — *Initialises the 7081*

140 WAIT 3000 — *Defines a waiting time to enable the 7081 to complete initialisation.*

150 GOSUB Setremote — *Switches the program to the 'Setremote' subroutine.*

160 ON INT #7 GOSUB Intserve! DEFINE INTERRUPT SERVICE ROUTINE

*When the 7081 interrupts the controller the program switches to the 'Intserve' subroutine.*

170 CONTROL MASK 7; 128! UNMASK GP-IB INTERRUPT

*Unmasks the GP-IB Interrupt bit.*

180 CARD ENABLE 7! ENABLE INTERRUPT

*Enables the interrupt card in the controller.*

190 OUTPUT 716; "HISTORY, EXPAND"

*Sets the 7081 History file to expanded format, i.e. 500 fully formatted readings.*

200 OUTPUT 716; "NINES = 5" — *Defines the 7081 scale length and integration time. At power-up the voltmeter adopts mode = Vdc, range = Auto so these commands do not have to be specified.*

210 OUTPUT 716; "BEGIN = 0,0,10"

*Sets up the start time of the clock to 10 seconds after activation of clock control.*

220 OUTPUT 716; "INTERVAL = 0,0,10"

*Defines the interval between measurements as 10 seconds.*

230 OUTPUT 716; "END = 0,1,30: CLOCK = ELAPSED"

*Sets the end time as one minute, 30 seconds after the start and defines the clock time as elapsed time as opposed to real time.*

240 OUTPUT 716; "MEASURE, CLOCK CONTROLLED "

*Activates measurements under the control of the clock set up in lines 210, 220 and 230.*

```
250 OUTPUT 716; "SRQ, READY, ON"
```
*Enables the voltmeter Service Request bit and defines it to be asserted when 7081 is ready, i.e. not busy.*

```
260 GOSUB Waitsignal
```
*Switches the program to the 'Waitsignal' subroutine.*

```
270 OUTPUT 716; "SRQ,OFF"
```
*Disables the voltmeter Service Request bit.*

```
280 OUTPUT 716; "LIMITS, ON"
```
*Turns on the 7081 Limits program*

```
290 OUTPUT 716; "COMPUTE, HISTORY"
```
*Sends the contents of the History file, i.e. the clock controlled measurements, through the Limits program.*

```
300 OUTPUT 716; " SRQ, READY, ON"
```
*Enables the voltmeter Service Request bit and defines it to be asserted when 7081 is ready, i.e. not busy.*

```
310 GOSUB Waitsignal
```
*Switches the program to the 'Waitsignal' subroutine.*

```
320 OUTPUT 716; "SRQ, OFF: OUTPUT, GP-IB, ON"
```
*Disables the voltmeter Service Request bit and turns the GP-IB output on.*

```
330 OUTPUT 716; "LIMITS, PEAK TO PEAK?"
```
*Asks for the peak to peak value from the Limits program.*

```
340 ENTER 716; A$
```
*Enters the result into A$.*

```
350 PRINT A$
```
*Prints out the result from A$.*

```
355 DISP "EXAMPLE COMPLETE"
```
*Prompts the controller to display 'Example complete.'*

```
360 STOP
```
*Program terminator.*

```
370 Waitsignal: Waiting = 1! SET WAITING FLAG
```
*This line supplies both a label for the 'Waitsignal' subroutine, i.e. marks the beginning, and sets the Waiting flag to 1.*

```
380 Idle = 0
```
*Sets the controller Idle flag to zero.*

```
390 Repeatwait : ! REPEAT
```
*Defines a program label to return to.*

```
400 Idle = Idle + 1
```
*Increments the Idle flag.*

```
410 DISP "STATUS : WAITING FOR INTERRUPT" ; Idle
```
*Prompts the controller to display 'Status: Waiting for Interrupt' followed by the value of the Idle flag.*

```
420 IF Signal = 0 THEN GOTO Repeatwait! REPEAT UNTIL SIGNAL SET
```
*Switches the program back to the line labelled 'Repeatwait' unless the Interrupt Acknowledge flag has the value 1.*

```
430 Signal = 0
```
*Resets the Interrupt Acknowledge flag.*

```
440 PRINT " "
```
*Prompts the controller to print out a blank line.*

```
450 RETURN
```
*Switches the program back to the line after the subroutine was called i.e. lines 270 and 320.*

```
460! INTERRUPT SERVICE ROUTNE
```
*This line serves no programming function. It simply serves as a title used to describe the subroutine which follows. The Interrupt Service routine in this program is general purpose as it shows how to access all of the main 7081 SRQ features.*

```
470 Intserve : !
```
*Label line which marks the beginning of the Interrupt Service routine.*

```
480 Int = Int + 1
```
*Defines the number of the Interrupt.*

```
490 PRINT "INTERRUPT#" ; Int
```
*Prompts the controller to print out 'Interrupt #' followed by the number of the interrupt.*

```
500 PPOLL CONFIGURE 716; "00001011" ! BIT 3, SENSE TRUE
```
*Assigns Bit 3 sense true to the 7081.*

```
510 P = P POLL (7)! CONDUCT POLL
```
*Causes the controller to perform a parallel poll.*

```
520 GOSUB Remotelocal
```
*Switches the program to the 'Remotelocal' subroutine.*

```
530 IF BIT (P,3) <> 1 THEN GOTO Endpol ! 7081 REQUESTING SERVICE?
```
*Switches the program to the end of the polling sequence if bit 3 is not true during a parallel poll.*

```
540 STATUS 716; S
```
*The controller conducts a serial poll.*

```
550 IF BIT (S,Rqs)<> 1 THEN GOTO Endpol
```
*Switches the program to the end of the polling sequence if the Request for Service bit is not set to 1 during a serial poll.*

```
560 IF BIT (S,Abn)<> 1 THEN GOTO Tryrdy! CASE ABNORMAL:
```
*Switches the program to the SRQ on Ready sequence if the Abnormal bit is not set to 1 during a serial poll.*

```
570 Abncode = BINAND (S, Abnmask)
```
*Performs a binary AND of the Serial Poll Byte and the Abnormal Code Mask bit enabling the reading of the Abnormality Code bits.*

```
580 ON Abncode + 1 GOTO Comint, Comex, Cal, Iovf
```
*On Abnormality Code + 1 the program switches to the lines labelled 'Comint', 'Conmex', 'Cal' or 'Iovf'.*

```
590 Comint : PRINT "ABNORMAL SET ★★★ COMMAND SYNTAX ERROR"
```
*Prompts the controller to print out 'Abnormal Set ★★★ Command Syntax Error'*

600 GOTO Tryrdy

*Switches the program to the line labelled 'Tryrdy'.*

610 Comex: PRINT "ABNORMAL SET ★★★ COMMAND EXECUTION ERROR"

*Prompts the controller to print out 'Abnormal Set ★★★ Command Execution Error.'*

620 GOTO Tryrdy

*Switches the program to the line labelled 'Tryrdy'.*

630 Cal : PRINT "ABNORMAL SET ★★★ CALIBRATION FAULT"

*Prompts the controller to print out 'Abnormal Set ★★★ Calibration Fault'.*

640 GOTO Tryrdy

*Switches the program to the line labelled 'Tryrdy'.*

650 Iovf: PRINT "ABNORMAL SET ★★★ INPUT BUFFER OVERFLOW"

*Prompts the controller to print out 'Abnormal Set ★★★ Input Buffer Overflow'.*

660 Tryrdy:!

*Line label signifying the beginning of the Service Request on Ready sequence.*

670 IF BIT (S,Rdy)=1 THEN GOTO Tryout

*Switches the program to the SRQ on Output Available sequence if the Ready bit is set to 1 during a serial poll.*

680 PRINT "READY"

*Prompts the controller to print out 'Ready'.*

660 Tryout:!

*Line label signifying the beginning of the SRQ on Output Available sequence.*

700 IF BIT (S,Out) <>1 THEN GOTO Tryusr! OUTPUT AVAILABLE?

*Switches the program to the SRQ on User sequence if the Output Available bit is not set to 1 during a serial poll.*

710 PRINT "OUTPUT AVAILABLE"

*Prompts the controller to print out 'Output Available'.*

720 ENTER 716; A$

*Enters each measurement into A$.*

730 PRINT A$

*Prints out each measurement placed in A$.*

740 Tryusr:!

*Line label signifying the beginning of the Service Request on User sequence.*

750 IF BIT (S,Usr) <>1 THEN GOTO Endtry ! USER?

*Switches the program to the end of the SRQ sequence if the User bit is not set to 1 during a serial poll.*

760 PRINT "USER SRQ"

*Prompts the controller to print out 'User SRQ'.*

770 Endtry:!

*Line label signifying the end of the Interrupt sequence.*

780 IF Waiting = 0! THEN GOTO Endwait ! IF BACKGROUND WAITING

*Switches the program to the line labelled Endwait if the 'Waiting for Signal' flag is reset.*

790 Waiting = 0! THEN CLEAR WAITING FLAG

*Resets the Waiting flag if set.*

800 Signal = 1! SIGNAL BACKGROUND

*Sets the Interrupt Acknowledge flag to 1 to indicate that the interrupt has been attended to.*

810 Endwait:! *Line label signifying Waiting flag reset.*

820 Endpol:! *Line label signifying end of polling sequence.*

830 PPOLL UNCONFIGURE 716 *Unconfigures the parallel poll previously set up.*

840 GOSUB Remotelocal *Switches the program to the 'Remotelocal' subroutine.*

850 CARD ENABLE 7! RE-ENABLE INTERRUPT

*Re-enables the interrupt card in the controller.*

860 RETURN *Switches the program back to the line after the subroutine was called, i.e. line 170.*

870 Setremote : ! *Line label marking the beginning of the 'Setremote' subroutine.*

880 Remote = 1! SET REMOTE FLAG

*Sets the 'Remote' flag to 1.*

890 REMOTE 716 *Puts the 7081 into the remote state.*

900 LOCAL LOCKOUT 7 *Disables the 7081 'local' key.*

910 RETURN *Switches the program back to the line after the subroutine was called, i.e. lines 160 and 990.*

920 Setlocal:! *Line label marking the beginning of the 'Setlocal' subroutine.*

930 Remote = 0! CLEAR REMOTE FLAG

*Resets the Remote flag to zero.*

940 LOCAL 7 *Puts 7081 into the local state.*

950 RETURN *Switches the program back to the line after the subroutine was called, i.e. line 1010.*

960 Remotelocal:! *Line label marking the beginning of the 'Remotelocal' subroutine. This routine is needed because the HP9835A returns devices to local during a parallel poll.*

970 IF Remote = 0 THEN GOTO Elseremote ! IF STATE WAS REMOTE

*Switches the program to the line labelled Elseremote if the 7081 was previously in local.*

980 GOSUB Setremote! THEN RETURN TO REMOTE

*Switches the program to the 'Setremote' subroutine if the 7081 was previously in remote.*

990 GOTO Endremote ! *Switches the program to the line labelled Endremote.*

1000 Elseremote: GOSUB Setlocal! ELSE RETURN TO LOCAL

*Line labelled 'Elseremote' which switches the program to the 'Setlocal' subroutine.*

1010 Endremote:! *Line label signifying the end of the 'Remotelocal' subroutine.*

1020 RETURN *Switches the program back to the line after the subroutine was called, i.e. lines 530 and 850.*

# Chapter 3
# Command Language

**Contents**


| | | *Page No.* |
|---|---|---|
| 1 | Introduction | 3.2 |
| 2 | Commands | 3.2 |
| 3 | Command Execution Order | 3.49 |
| 4 | Error Messages | 3.50 |

This chapter is intended to familiarise the user with the commands used and error messages produced when the voltmeter is configured under remote control.

## 1 Introduction

The command language is for use with both the GP-IB and RS232 interfaces, and is made up of English language words, decimal numbers and a set of punctuation symbols.

The language symbols are defined in Table 3.1.

**Table 3.1** *Language Symbol Definitions*

| Symbols | Definition |
|---|---|
| MEAsure | Upper case characters indicate the required minimum abbreviation. |
| space | May be used anywhere to improve command readability. |
| ? | Causes the current status of the command to be output. |
| : | Used in multicommand lines to separate the commands, e.g. MODe = VDC:RANge = 100. |
| , | Used to separate command words. |
| = | May be used instead of ',' except where the context would be confused, e.g. SCale,M=2=C=4 will produce an error. |

## 2 Commands

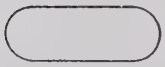
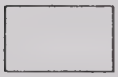
The command words available, with their page numbers, are given in Table 3.2.

Each command is represented by a syntax diagram. The symbols used in the diagrams are explained in Table 3.3

**Table 3.2** *Command Words*

| Command | Page No. | Command | Page No. |
|---|---|---|---|
| BEEp | 3.4 | INTerval | 3.26 |
| BEGin | 3.5 | LImits | 3.27 |
| CALIBRATE | 3.6 | LOck front panel | 3.30 |
| CAPitals lock | 3.7 | MEAsure | 3.31 |
| CHannel | 3.8 | MEMory | 3.33 |
| CLock | 3.9 | MODe | 3.34 |
| COmpute | 3.10 | NInes | 3.35 |
| DAte | 3.11 | NUll | 3.36 |
| DELAy | 3.12 | Output | 3.37 |
| DELImit | 3.13 | Pad count | 3.38 |
| DIGital filter | 3.14 | RANge | 3.39 |
| DISplay | 3.16 | RATio | 3.40 |
| DRift | 3.17 | SCale | 3.41 |
| DUmp | 3.18 | SRq | 3.42 |
| ENd | 3.19 | STATistics | 3.43 |
| ERror | 3.20 | STOp | 3.45 |
| FORmat | 3.21 | TEst | 3.46 |
| HElp | 3.23 | TIme | 3.47 |
| HIstory | 3.24 | TRigger | 3.48 |
| INItialise | 3.25 | | |

**Table 3.3** *Sytax Diagram Symbols*

| Symbols | Definition |
|---|---|
|  | Ovals are used to represent command words/abbreviations that must be entered by the user. Alternatives are shown inside the same symbol. |
| | Circles represent separators. Alternatives are shown inside the same symbol. |
|  | Rectangles contain values or elements that are to be defined or that are shown in their own diagram. |
|  | Lines and arrows indicate authorised paths and are used to show the acceptable sequences(s) of elements in the syntax diagram. |

Each command line should be terminated with either Carriage Return (RS232) or, Line Feed or EOI (GP-IB).

## BEEp

Enables a tone which can be used to draw the user's attention to a command and/or result.

**Fig 3.1** *Beep Command*

## BEGin

Used to set the start time for clock controlled measurements.

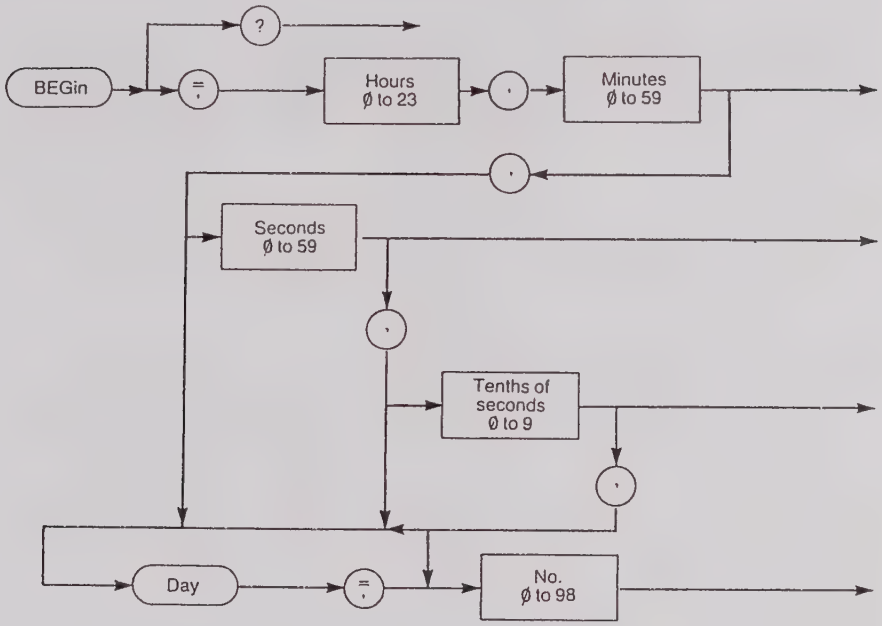



**Fig 3.2** *Begin Command*

The time should be entered in hours, minutes, seconds, tenths of seconds format. A day value can also be entered; this value indicates 'elapsed' days to the activation of clock control.

The Begin command is used in conjunction with the Interval, End and Clock commands. To initiate clock controlled measurements the Measure, Clock controlled command should be used.

**Examples**

| | |
|---|---|
| BEGin = 10,30 | – starts clock controlled measurements at 10.30 |
| BEGin = 15,45,Day=4 | – starts clock controlled measurements at 3.45pm in 4 days time |
| BEGin? | – produces a reply of the form: Begin = 15,45,11.5,Day=04 |

## CALIBRATE

Used to calibrate the unit against known standards.

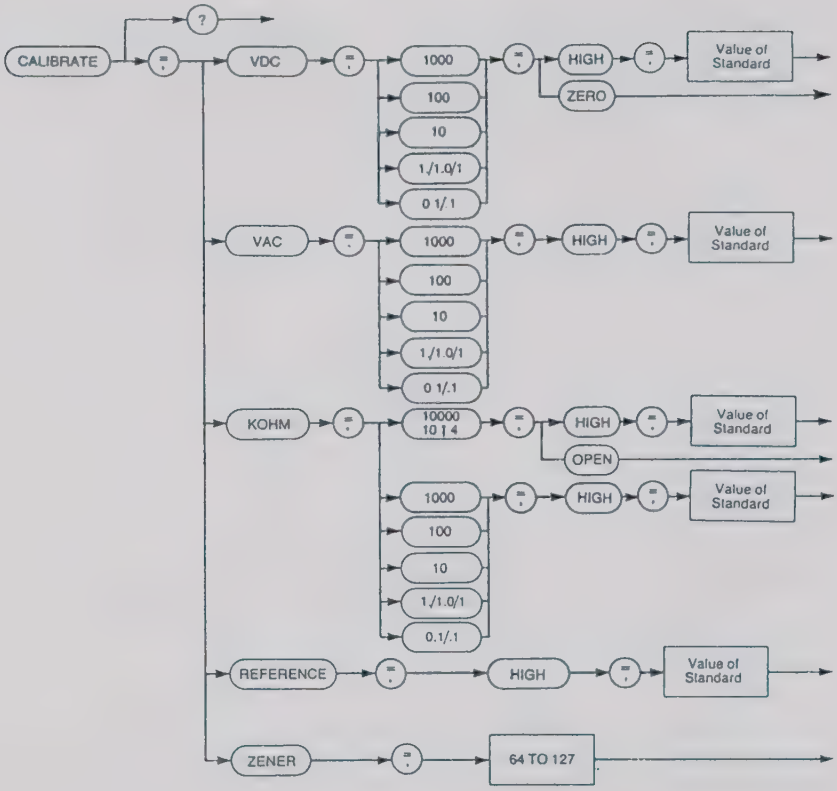



**Fig 3.3** *Calibrate Command*

The Calibrate command is only valid when the 7081 front panel key is turned to the CAL position.

**Examples**

CALIBRATE,
VDC=10,HIGH=10.000012 – high point calibration using a standard of known value equal to 10.000012 volts.

CALIBRATE? – produces a status reply of the form:

Calibrate = Fail 1
Fail 2
Fail 3
OK

Refer to the 7081 Maintenance Manual for an explanation of these messages.

### CAPitals lock

Used to set the output format to capitals only.

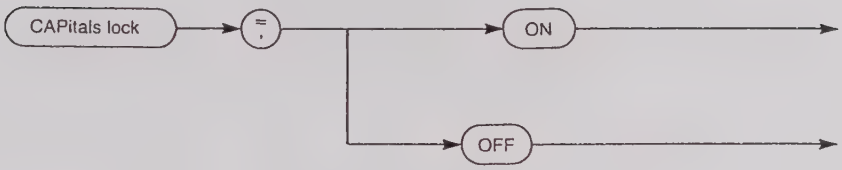



**Fig 3.4** *Capitals Lock Command*

This command can be set to on or off. If CAPS lock is on, all output messages have lower case characters converted to upper case and any commas or colons replaced by spaces.

**Example**

CAPitals lock = OFF – default condition.

## CHannel

Used to select a specific Minate channel.

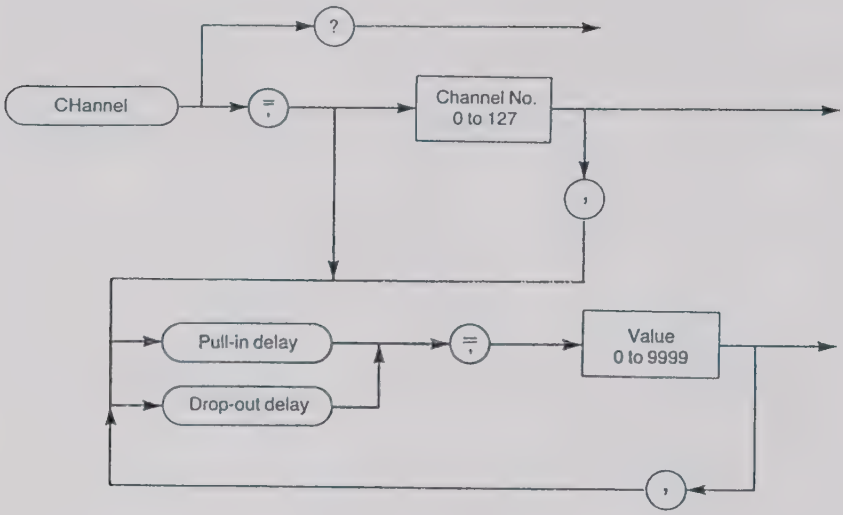



**Fig 3.5** *Channel Command*

This command selects a channel, for measurement purposes, and sets any pull-in or drop-out delays that may be required. Only one pull-in and one drop-out delay is available for all channels.

**Examples**

CHannel=40 – selects channel 40.

CHannel=12,Pull-in delay = 10,Drop-out delay = 5
– selects channel 12 and sets a pull-in delay of 10ms and a drop-out delay of 5ms.

CHannel? – produces a reply of the form:

Channel=12, Pull-in Delay = 10ms, Drop-out Delay = 5ms.

## CLock

Sets the mode of the clock control parameters.

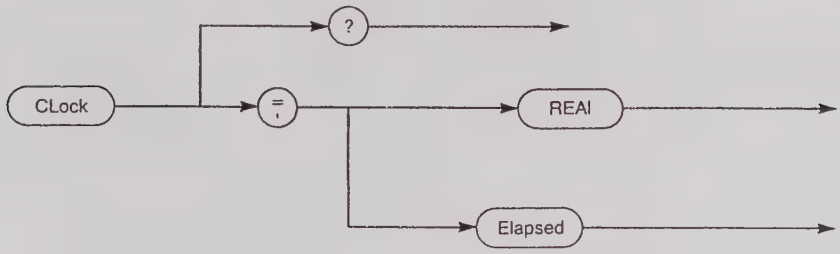



**Fig 3.6** *Clock Command*

This command is used in conjunction with the Begin, Interval and End commands. When Clock is set to real, the Begin and End values must be set to a time of day. When Clock is set to elapsed, the Begin time should be set to the value required between the moment a measurement is 'triggered' and the moment the first measurement is taken. The End value should indicate the period over which the voltmeter will continue to take measurements and the Interval value the time between individual measurements.

**Examples**

CLock=REAl – Begin, Interval and End times related to real time.

CLock=Elapsed – Begin, Interval and End times related to the time elapsed since activation.

CLock? – produces a reply of the form:

Begin = 12,34,56.7,Day=01
Interval = 00,00,30.0,Day=00
End = 12,54,56.7,Day=02
Clock = Real

## COmpute

This command is associated with the unit's programs.

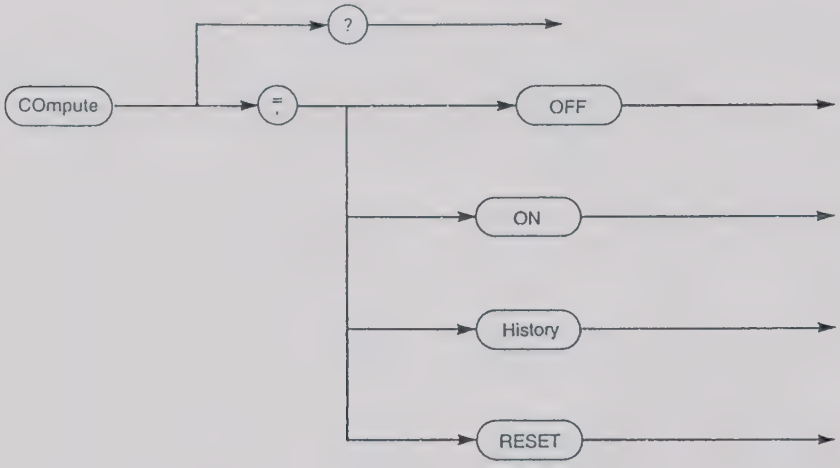



**Fig 3.7** *Compute Command*

Compute is automatically turned on, if a program is enabled, or off, if no programs are left on. The command can also be used to process the contents of the History file or reset the program running variables.

**Examples**

| | |
|---|---|
| COmpute=ON | – passes measurements through any activated programs in the chaining order. |
| COmpute=OFF | – measurements are not processed by the programs. |
| COmpute=History | – passes the contents of the History file through any activated programs in the chaining order and returns the processed results to the History file |
| COmpute=RESET | – resets all the program running variables whether activated or not. |
| COmpute? | – produces a reply of the form: |

Compute = OFF
Ratio = OFF
Digital Filter = OFF
Scale = OFF
Statisitics = OFF
Limits = OFF

## DAte

Enables setting of the calendar date into the unit.

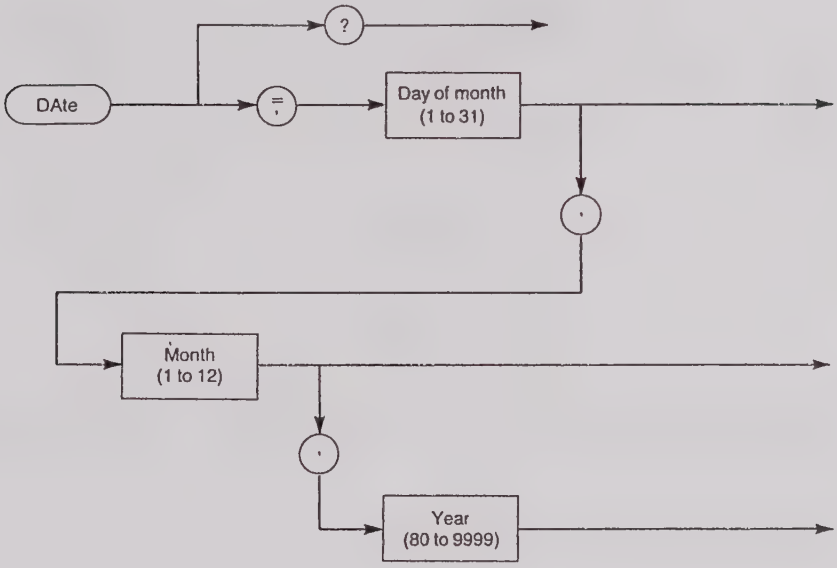



**Fig 3.8** *Date Command*

**Examples**

DAte=21,6,1983 – enters the date 21,6,1983 into the unit.

DAte? – produces a reply of the form:

Date = 21,6,1983

## DELAy

Defines the trigger delay.

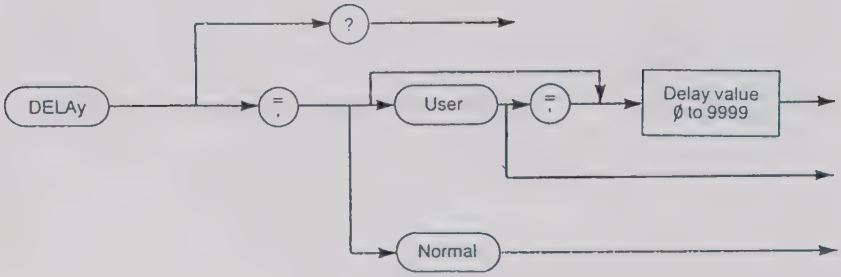



**Fig 3.9** *Delay Command*

This delay can be set to normal or a user defined time, entered in milliseconds, up to a maximum of 9999ms. If the delay is set to normal, the unit selects its own delay before carrying out a measurement.

**Examples**

DELAy=User,400 – trigger delay user defined at 400ms.

DELAy? – produces a reply of the form:

Delay = Normal
or
Delay = User,400ms.

## DELImit

Defines the end of line character.

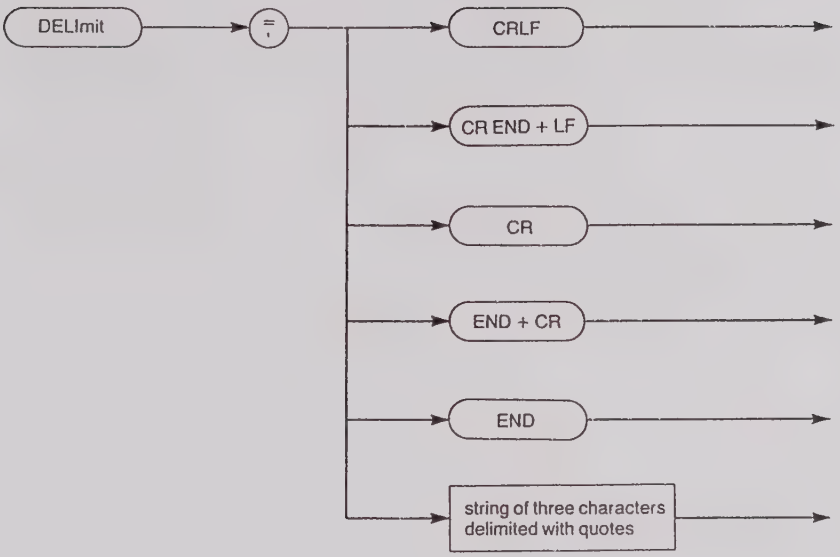



**Fig 3.19** *Delimit Command*

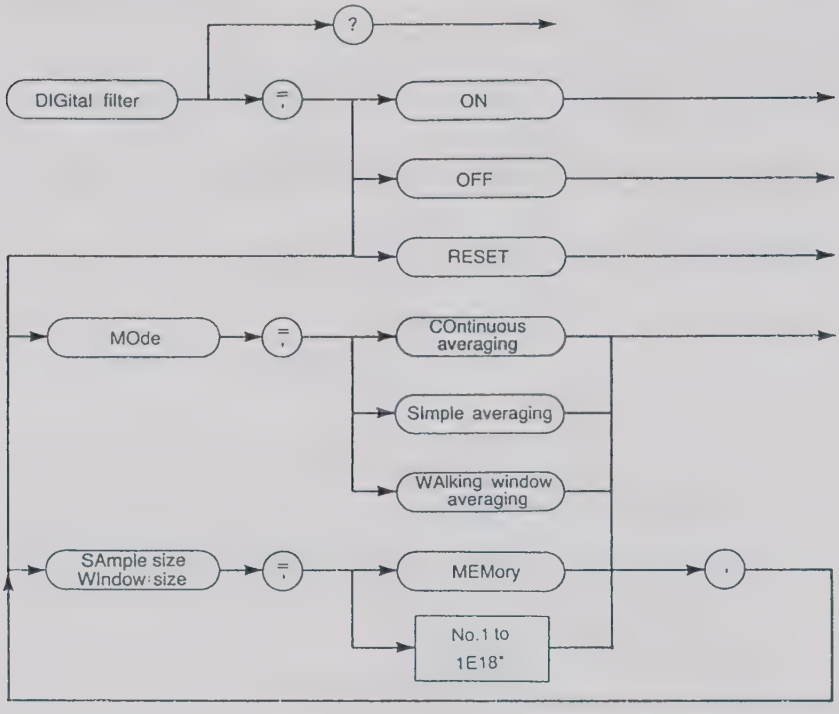
**Note:** END is defined as:
(a) assert EOI with the last character on the GP-IB.
(b) output ETX as the last character on RS232.

**Example**

DELImit=END+CR – sets the end of line character to END+CR

## DIGital filter

Enables selection and definition of the Digital Filter program.




* *If Mode = Walking window, the maximum window size is 16. If values greater than 16 are entered, the program will default to 16. No error message is produced to indicate that this has occurred and the response to DIGital filter? will give the value entered not the value used by the program*

**Fig 3.11** *Digital Filter Command*

This command can be used to set up all the program's parameters at once or just one parameter.

The program has three modes of operation:

Continuous averaging
Simple averaging
Walking window averaging

In continuous averaging, an updated result is output for every input whereas, in simple averaging, one average result is produced for each sample size. The sample size has a maximum value of 1E18.

In walking window averaging, the sample window used to produce the result consists of the last n readings where n is the defined sample size. Once the sample size has been fulfilled an average result is produced for every new input. The maximum sample size is 16.

The program adopts a default setting on initialise of walking window averaging with a sample size of 10

**Examples**

DIGital filter,MOde = WAlking window, WIndow size = 10,ON

– sets up the program to walking window mode with a window size of 10. This command also turns the program on.

DIGital filter,MOde = SImple averaging

– changes the program mode to simple averaging.

DIGital filter?

– produces a reply of the form:

Digital Filter = OFF
Mode = Walking Window Average
Window Size = 10.0000000E+00

## DISplay

Affects the voltmeter front panel display.

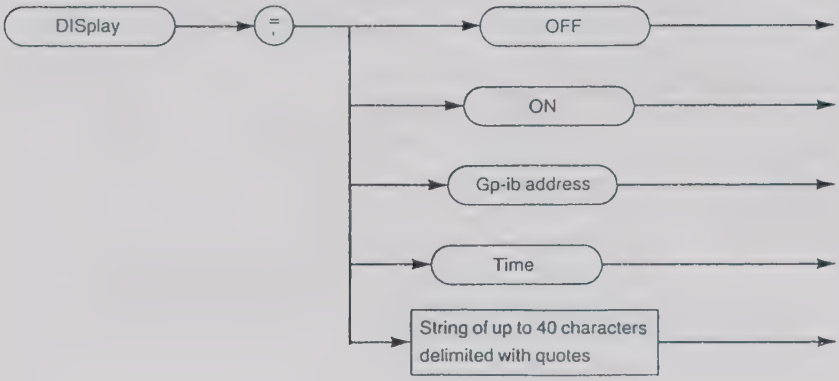



**Fig 3.12** *Display Command*

If a controller uses quotes (") for delimiting character strings, the quotes of the display command can be sent as their ASCII numeric value, the sending string being made up of these character values and the string itself, e.g.

OUTPUT 716; "DISPLAY,"&chr$(34)&"Hello"&chr$(34)

**Examples**

DISplay=Time – causes time of day to be continually displayed until displaced by other display information

DISplay,"1.23456789" – causes 1.23456789 to appear in the display until displaced by other display information.

## DRift

Enables or disables the automatic drift correct measurement.

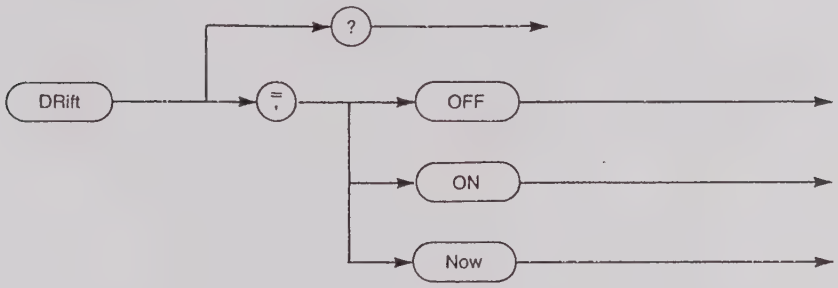



**Fig 3.13** *Drift Command*

A drift correct measurement is performed approximately once every 15 minutes and is also automatically invoked when the nines parameter is increased above 5 × 9. The measurement is performed at the number of nines integration selected or 6 × 9 whichever is the greater.

**Examples**

| | |
|---|---|
| DRift = Now | – enables an immediate drift correct. |
| DRift = OFF | – disables automatic drift correct. (A drift correct measurement will still be performed if the nines parameter is increased above 5 × 9). |
| DRift? | – produces a reply of the form:<br><br>Drift Correct = ON. |

## DUmp

Enables the output of the contents of the specified History file locations.

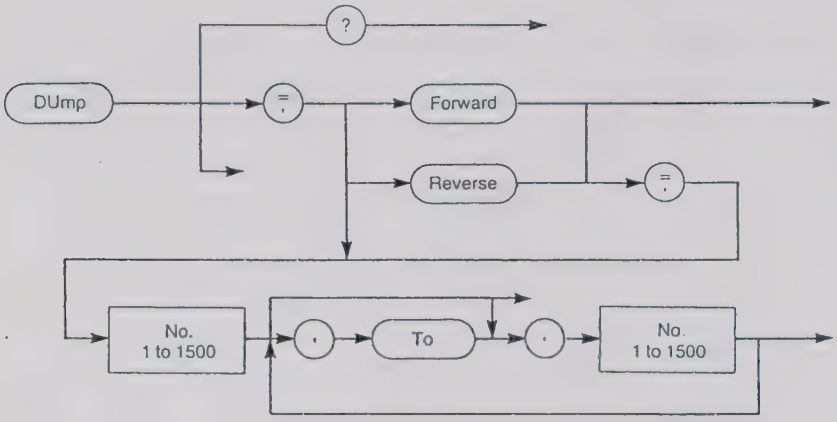



**Fig 3.13** *Dump Command*

The History contents are preserved after a Dump command. Forward direction defines the oldest record as history record number 1; reverse direction defines the newest record as history record number 1.

**Examples**

DUmp – outputs all the history records in the direction last specified. The default direction is forward. If no history exists, an error message is output.

DUmp=1,To,20,25 – outputs history records 1 to 20 and 25.

DUmp=Reverse,20,To,25,30,To,26 – outputs history records 20 to 25, 30 to 26 relative to the newest record.

DUmp? – produces a reply of the form:

Dump Direction = Forward, nnnn
where nnnn is the number of records present.

## ENd

Used to set the end time for clock controlled measurements.

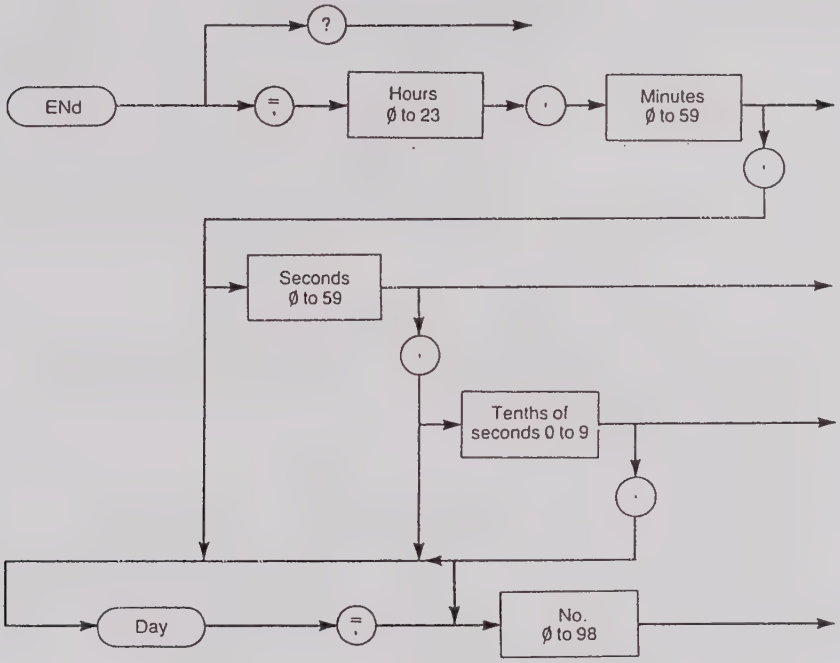



**Fig 3.15** *End Command*

The time should be entered in hours, minutes, seconds, tenths of second format. A day value can also be entered : this value indicates 'elapsed' days since the start of clock control.

The End command is used in conjunction with the Begin, Interval and Clock commands. To initiate clock controlled measurements the Measure, Clock controlled command should be used.

**Examples**

ENd=15,45, Day=2 – ends clock controlled measurements at 3.45pm, 2 days after the start.

ENd? – produces a reply of the form:

End = 15,45,11.5,Day=02

## ERror

Defines the type of error messages that are produced.

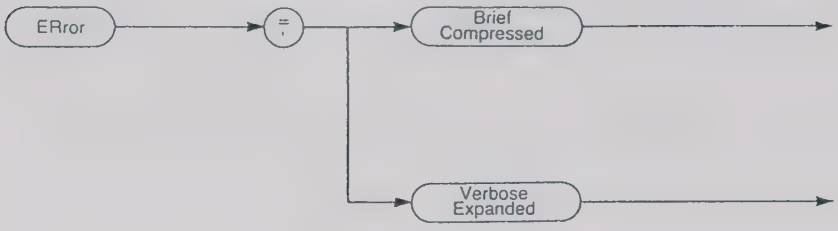



**Fig 3.15** *Error Command*

The messages displayed are as shown in Table 3.4. For a full explanation of the error messages refer to the end of this chapter.

**Table 3.4** *Error Messages*

| Command | Brief Message | Verbose |
|---|---|---|
| SCale,M=2,C=2 | OK | Command Syntax OK. |
| DUmp | E50 | No History Present. |
| SCale,M=2=C=2 | E4 | Invalid Separator Before Char No. 10<br>This Part:2= |

**Example**

ERror = Verbose – error messages are produced, instead of error numbers, on RS232 and at the GP-IB output if on

## FOrmat

Defines the nature of the voltmeter output.

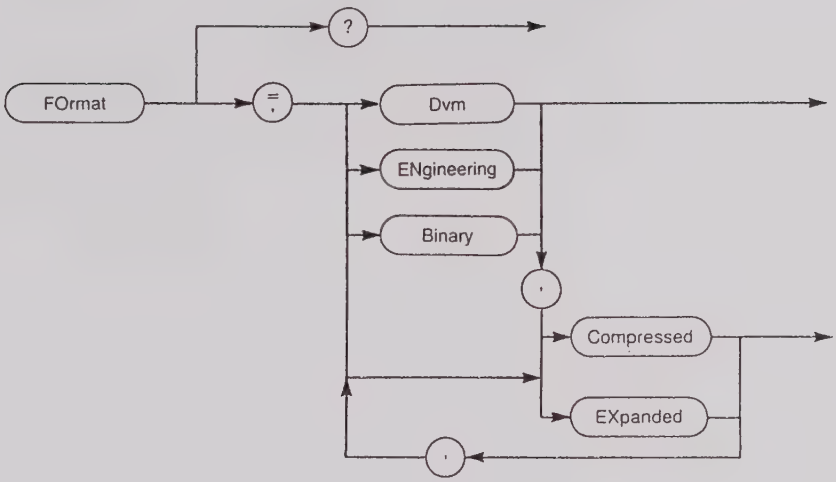



**Fig 3.17** *Format Command*

Three format settings are available: DVM, Engineering and Binary, all of which can be selected as Compressed or Expanded

DVM – the decimal point occupies a fixed position in the output and the number is arranged to give a positional indication of magnitude, e.g.

```
 -0.11234567
-1123.45678
-11234.56
```

If DVM, Expanded is selected, the units, time, day, channel and history file number (as applicable) will also be output, i.e. -1.52345 Vdc Time = 12,34,56.7,Day=01 Channel 123 Hist No:0123.

Engineering – the number occupies a fixed position but the decimal point may occupy one of three positions. The exponent, which is always shown, is allowed to change in intervals of three, e.g.

```
-112.3456E-06
  1.12345E+09
-11.2345678E-06
```

Engineering, Expanded produces an output of the form: 156.3445E+09 Vdc Time = 12,34,56.7, Day=00 Channel 123 Hist No:0034

Binary – the IEEE 488 recommended format for binary real numbers is used. Binary, Expanded gives a representation of the time, day, channel and history file number.

For further details on the formats available refer to Appendix 1 at the end of the manual

**Examples**

| | |
|---|---|
| FOrmat=Dvm,EXpanded | – measurements displayed in DVM form with time, day, channel and history file number included. |
| FOrmat? | – produces a reply of the form:<br><br>Format=Expanded, DVM:Caps Lock = OFF |

### HElp

Explains the last error message, i.e. displays the verbose reply.

Fig 3.18 *Help Command*

## HIstory

Used to define the size and nature of the History file.

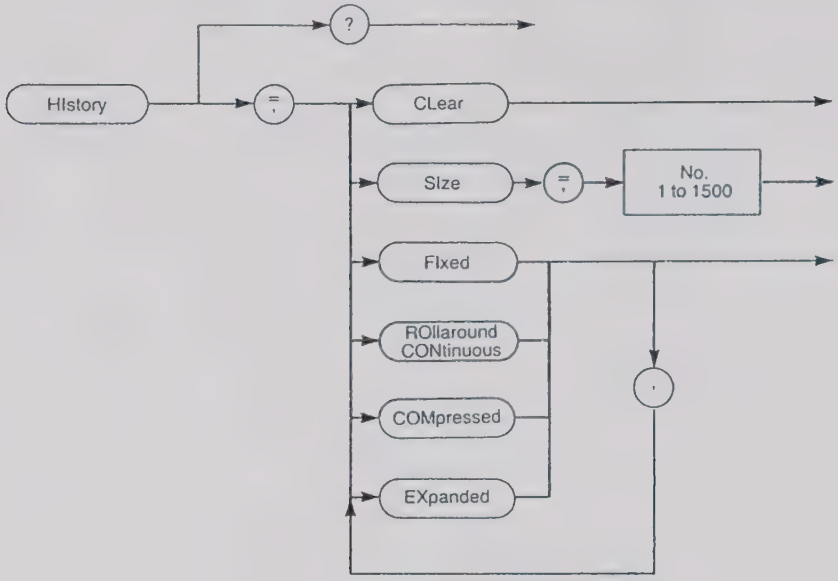



**Fig 3.19** *History Command*

In order to display the History file contents in expanded form, both History and Format must be set to Expanded. If Format is set to Compressed, the History file contents will be stored in expanded format but output in compressed format.

**Note:** History size values >500 are not considered an error even if History is set to Expanded. In this instance, the file will default to 500.

**Examples**

| | |
|---|---|
| HIstory,FIxed,SIze=100 | – In fixed, the History file is not overwritten, hence the next 100 results are saved. The voltmeter continues measuring after the History file is full but no further results are stored. |
| HIstory,ROllaround, COMpressed, SIze=1500 | – In rollaround the last n readings are maintained in the file where n is the history size specified. In compressed, only the numeric values of up to a maximum of 1500 results are stored. |
| HIstory,EXpanded | – In expanded, full result information of up to a maximum of 500 results are stored |
| HIstory? | – produces a reply of the form:<br>History,Compressed,Roll,Size=500. |

### INItialise

Causes the unit to return to the initialised state.

**Fig 3.2** *Initialise Command*

The GP-IB status and handshake are not preserved during an initialise execution.

## INTerval

Used to set the interval time for clock controlled measurements.

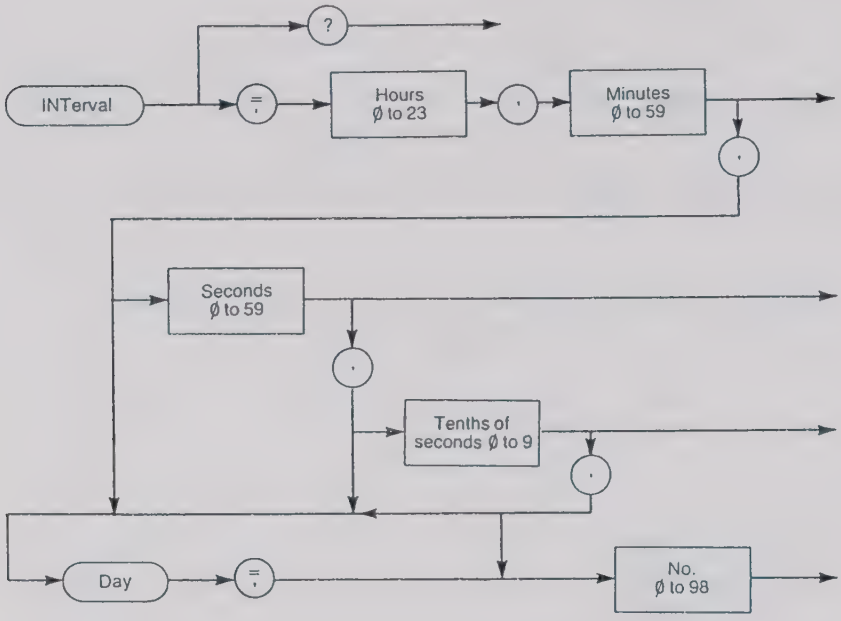



**Fig 3.21** *Interval Command*

The time should be entered in hours, minutes, seconds, tenths of seconds format. A day value can also be entered: this value indicates 'elapsed' days.

The Interval command is used in conjunction with the Begin, End and Clock commands. To initialise clock controlled measurements the Measure, Clock controlled command should be used.

**Examples**

INTerval=0,8,Day=0 – sets the interval time to 8 minutes.

INTerval? – produces a reply of the form:

Interval =00,08,00.0,Day=00

## LImits

Enables selection and definition of the Limits program

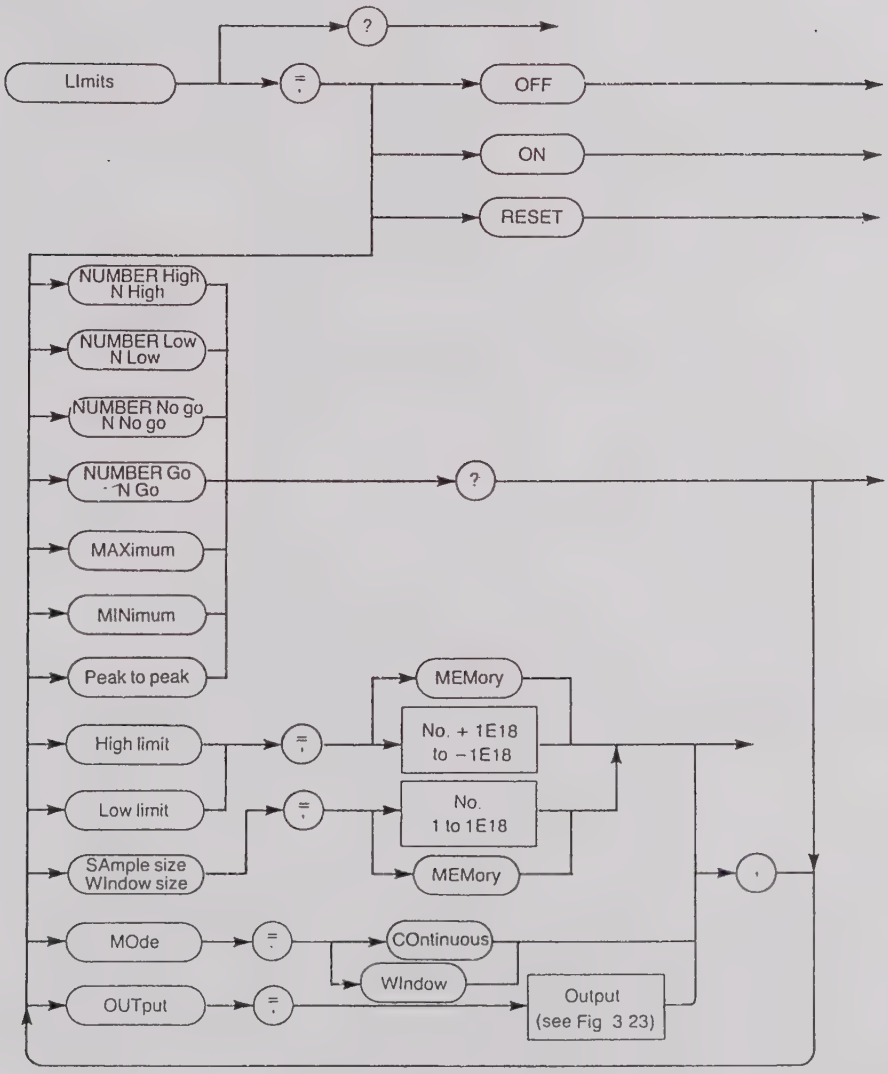



Fig 3.22 *Limits Command*

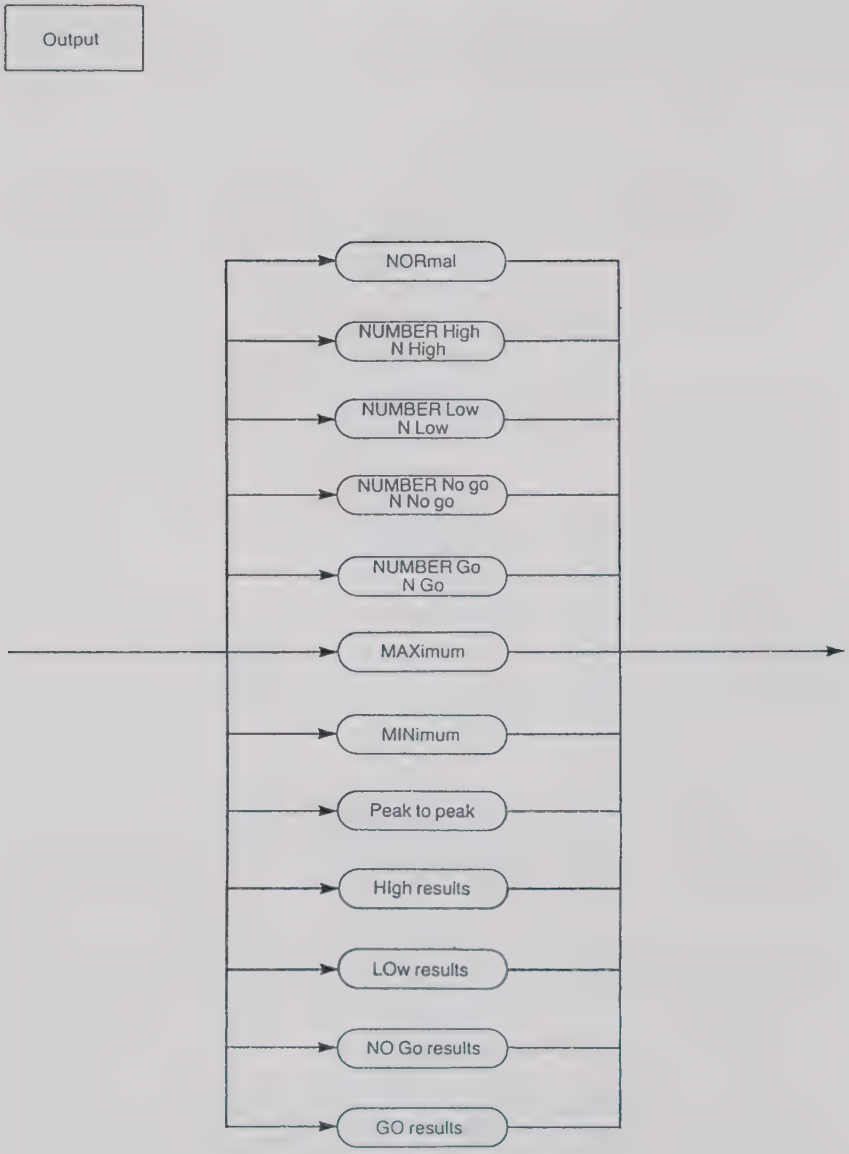



**Fig 3.23** *Limits Output Command*

The Limits command can be used to set up all of the program functions at one time, a particular function of the program or query a result.

Two program modes are available:

Continuous sampling
Window sampling

In continuous sampling, the results are constantly updated and an output result is displayed for every input.

In window sampling, one result is produced for each sample size. The maximum sample size is 1E18.

The output of the program may be configured to be any of the following:-

Normal (reading passed on unaltered)
Number high (number above high limit)
Number low (number below low limit)
Number no go (number out of limits)
Number go (number within limits)
Max. value
Min. value
Peak to peak value
High results (only high results output)
Low results (only low results output)
No go results (only out of limits results output)
Go results (only within limits results output)

The 7081 stores the number of high, low, go and no go results, the max., min., and peak to peak values in the program memory. These results can be recalled at any time during the program's life by entering a query command, e.g.

LImits,MAXimum?

**Examples**

| | |
|---|---|
| LImits,MOde=WIndow, SAmple size=10,High limit=6, Low limit=3,OUTput=GO results,ON | – sets up the program for window sampling with a sample size of 10, high limit of 6, low limit of 3, output results within limits. The command also turns the program on. |
| LImits,OUTput=MAXimum, ON | – sets the program output to maximum and turns the program on. |
| LImits,Peak to peak? | – produces a reply of the form:<br><br>P TO P = 1.0000000E+00 |
| LImits? | – produces a reply of the form:<br><br>Limits = ON<br>Mode = Window Sampling<br>Output = Max.<br>Hi Limit = 6.0000000E+00<br>Lo Limit = 3.0000000E+00<br>Sample Size = 10.0000000E+00 |

### LOck front panel

Enables or disables the front panel controls.

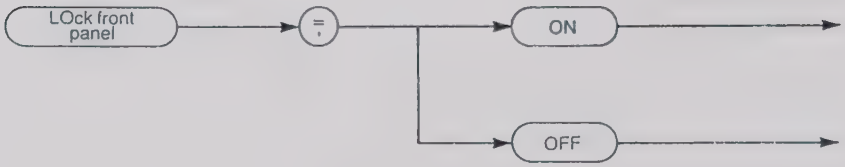



Fig 3.24 *Lock Front Panel Command*

The front panel controls are enabled at power-up.

**Examples**

LOck front panel = ON – front panel controls disabled.

LOck front panel = OFF – front panel controls enabled.

## MEAsure

Used to initiate the taking of measurements.

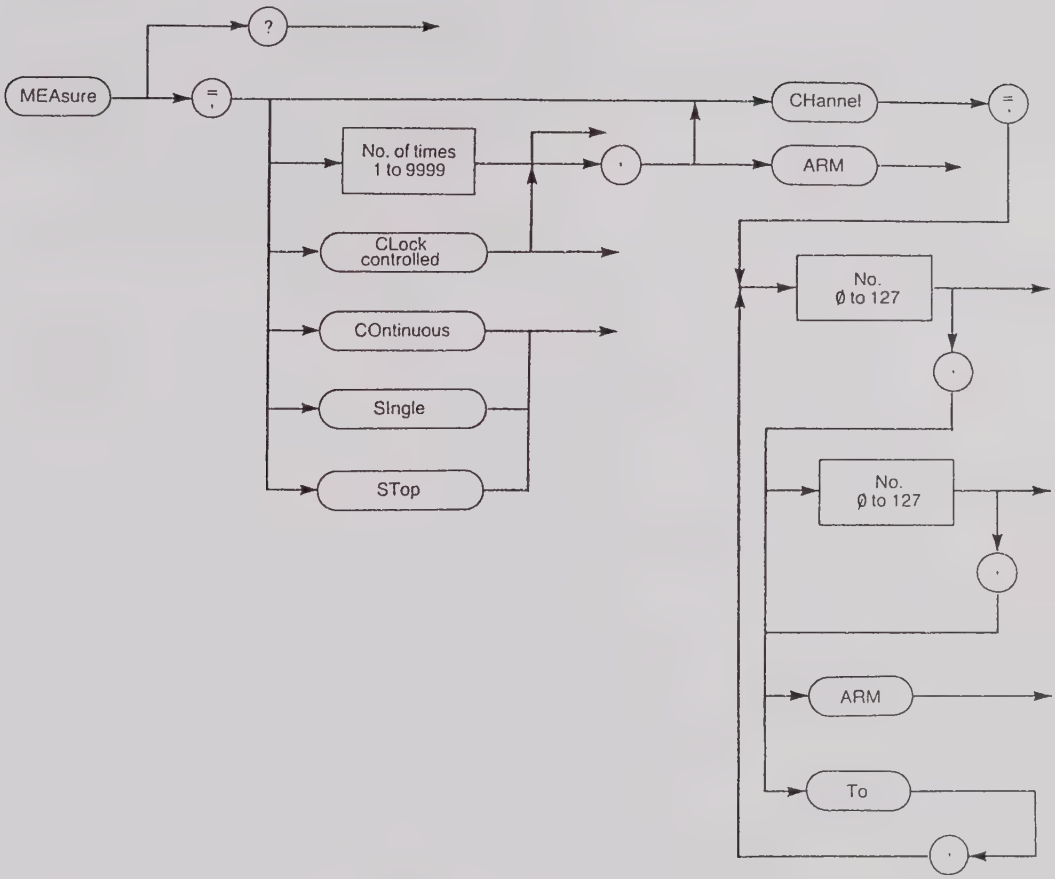



**Fig 3.25** *Measure Command*

Five main settings are available for the command: Clock controlled, Continuous, Single, Stop and a specified number from 1 to 9999. The channel can also be specified. If the word 'ARM' is used in the command, the voltmeter will wait for a trigger signal before proceeding with the measurements.

**Examples**

MEAsure,CONtinuous – initiates measurements at the fastest possible rate

MEAsure,STop – this command can be used to stop continuous measurements. After the command has been entered, one result will appear and then the measuring will cease. If this command is used to stop any of the other measure commands, the unit will only cease measuring after the previous command has been implemented.

e.g. MEAsure,25
MEAsure,STop

The unit will take 25 measurements and then stop.

MEAsure,SIngle – initiates a single measurement.

MEAsure,CHannel,15,To,23,9 – initiates measurements on channels 15 to 23, and 9

MEAsure,25 – initiates 25 measurements.

MEAsure,CLock controlled, CHannel,23,To,45
– initiates measurements, under clock control, of channels 23 to 45, i.e. each event under clock control is a scan of channels.

MEAsure,5,ARM – initiates 5 measurements on receipt of a trigger signal.

MEAsure? – produces a reply of the form:

Measure = Stop
or
Measure = Continuous

## MEMory

Used to store items in or recall items from the memory.

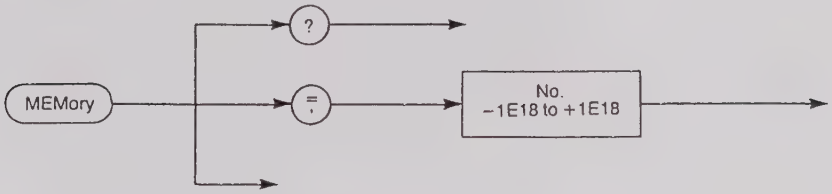



**Fig 3.26** *Memory Command*

**Examples**

MEMory – stores the last numeric output.

MEMory,3.56 – stores 3.56 in the memory.

MEMory? – recalls the memory contents, i.e.

Memory Contents = 3.5600000E+00

## MODe

Enables selection of the measurement function.

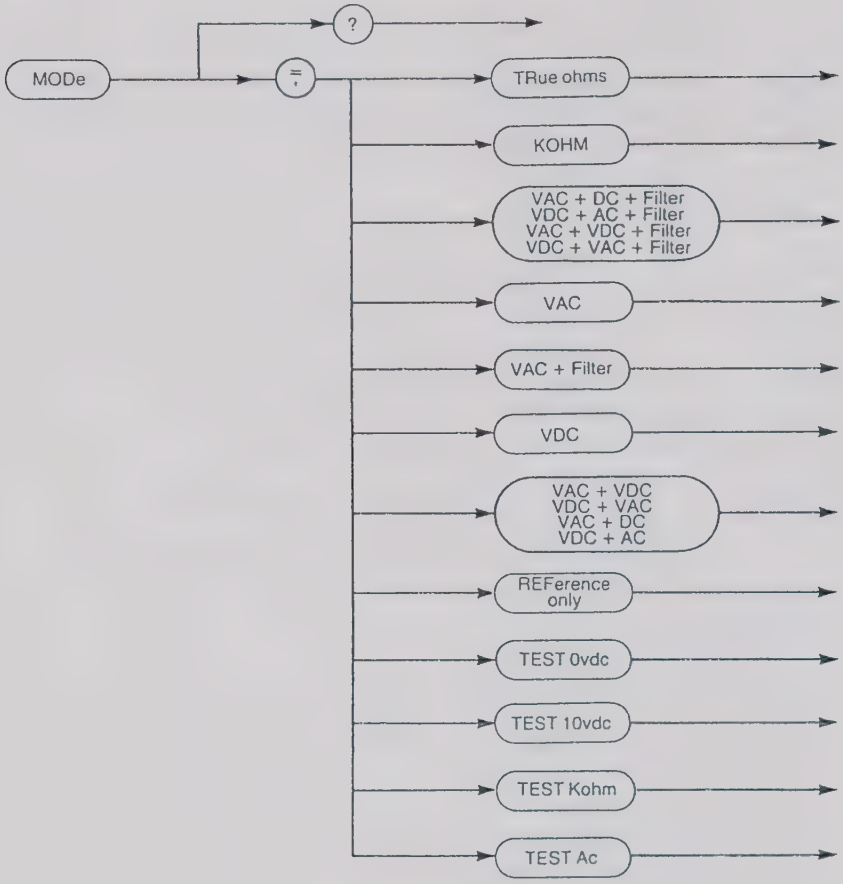



**Fig 3.27** *Mode Command*

At power-up and initialise 7081 adopts Mode=VDC.

**Examples**

MODe=VAC – selects the ac voltage function

MODe=TRue ohms – selects the true resistance function.

MODe? – produces a reply of the form:

Mode = VDC [Front]
where [Front] identifies the input terminals enabled.

## NInes

Enables setting of the scale length to any value from 3 to 8 × 9 inclusive.

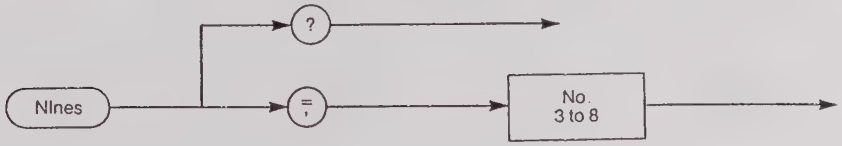



**Fig 3.28** *Nines Command*

**Examples**

NInes=7 – sets a scale length = 7 × 9

NInes? – produces a reply of the form:

Nines = 3 × 9's

## NUll

Enables or disables the present null values, or enables a new null measurement.

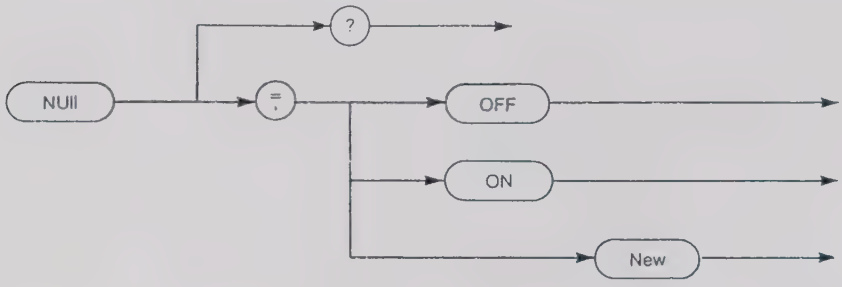



**Fig 3.29** *Null Command*

**Examples**

NUll,New – initiates a new null measurement. If RANge = Auto, all ranges for the unit mode are nulled. Each nullable mode has its own null value set.

NUll,ON – enables the present null value set.

NUll? – produces a reply of the form:

Null=ON

## Output

Enables selection of the unit output.

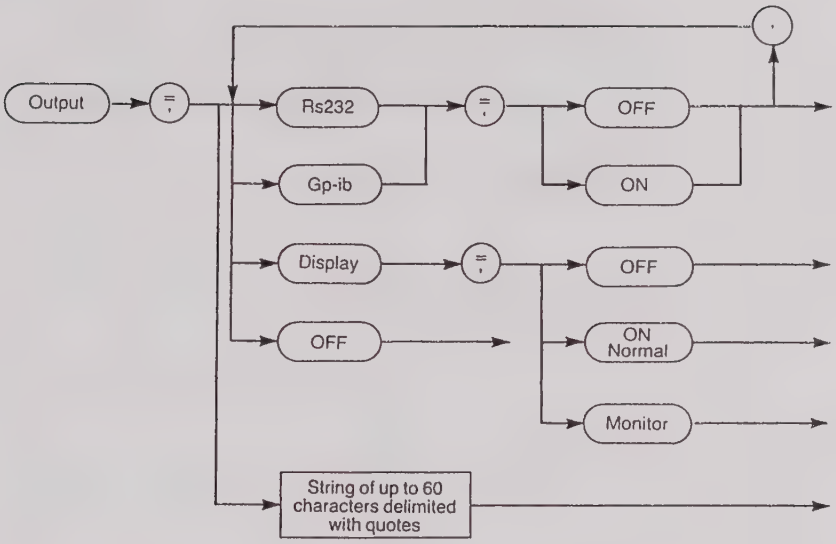



**Fig 3.30** *Output Command*

The RS232 and GP-IB interfaces, and the voltmeter display can be turned on or off. The voltmeter display can also be set to Normal or Monitor: with Monitor selected the display gives an indication of everything going out to the interfaces, e.g. error messages.

If the controller uses quotes (") for delimiting character strings, the quotes of the output command can be sent as their ASCII numeric value, the sending string being made up of these character values and the string itself e.g.

OUTPUT 716; "OUTPUT," & chrs$(34) & "Hello" & chrs$ (34)

**Examples**

| | |
|---|---|
| Output, Rs232, ON | – turns the RS232 output on. |
| Output,Display=Normal | – sets the voltmeter display to normal operation. |
| Output,OFF | – turns all interface outputs off. |

### Pad count

Used to set the number of nulls (0 to 9) after a Carriage Return and before a Line Feed at the RS232 output.

**Fig 3.31** *Pad Count Command*

**Example**

Pad count = 0 – no nulls inserted.

## RANge

Used to set the range used for taking measurements.

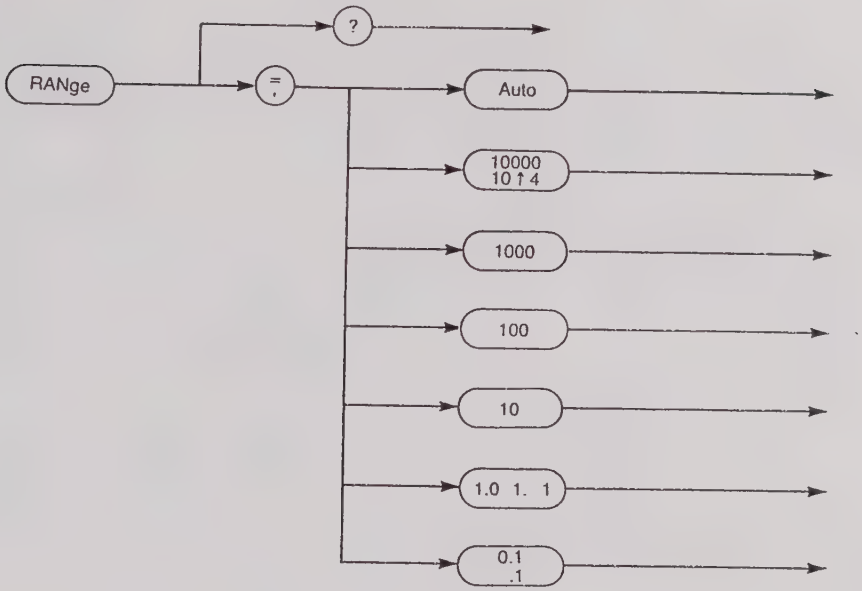



**Fig 3.32** *Range Command*

Not all ranges are available on all measurement functions but no error is reported if a range is selected which is not available on the current function. The unit remembers any attempted range selections for possible future implementation when a suitable function is selected.

**Examples**

RANge = 10000 – selects the 10000 range

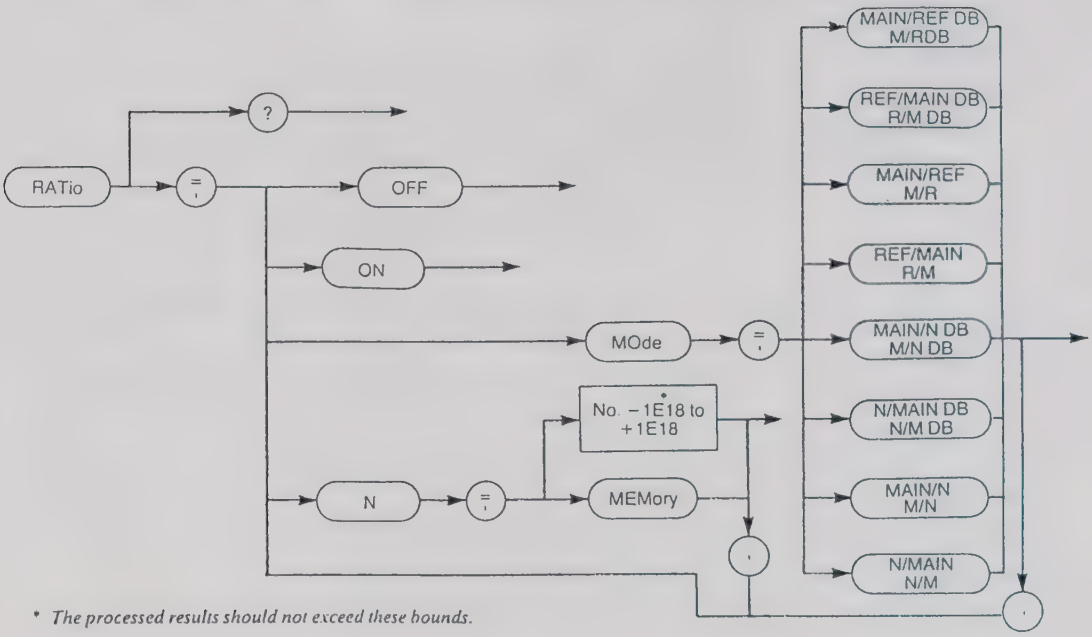
RANge = Auto – selects auto ranging

RANge? – produces a reply of the form:

Range = 100, Auto
or
Range = 100, Fixed

## RATio

Enables selection and definition of the Ratio program.




* *The processed results should not exceed these bounds.*

**Fig 3.33** *Ratio Command*

The Ratio command can be used to set up all of the program functions at one time or a particular function of the program

'Main' refers to the voltmeter input terminals (front or rear panel) and 'Reference' to the reference or ratio terminals (rear panel). N is a user defined constant.

The program default setting, which is adopted at initialise, is Main/Reference.

**Examples**

RATio,MOde=Main/Ref DB, ON

– provides results of the ratio of the main input terminals to the reference terminals and gives the values in dBs.

RATio, MOde=Main/N,N=5.6, ON

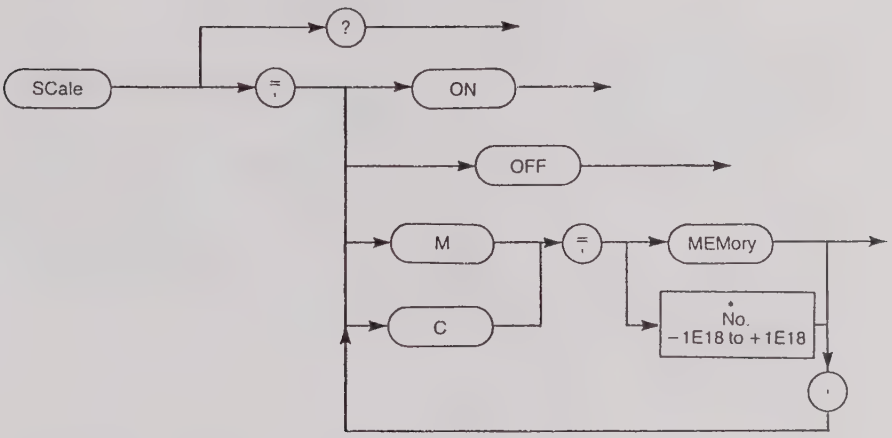
– provides results of the ratio of the main input terminals to a constant N (5.6).

RATio? – produces a reply of the form:

Ratio = OFF
Mode = Main/N
N=5.6000000E+00

## SCale

Enables selection and definition of the Scale program.




* *The processed results should not exceed these bounds.*

**Fig 3.34** *Scale Command*

The Scale program provides:

$y = mx + c$

where $y$ is the processed result, $x$ the input measurement, and $m$ and $c$ user defined constants. Both $m$ and $c$ may be taken from memory, thus enabling a previously measured value to be used.

**Examples**

SCale,M=28.1,C=MEMory,ON – turns the program on with M=28.1 and C=the contents of the memory.

SCale,OFF – turns the program off.

SCale? – produces a reply of the form:

Scale=On
M=28.1000000E+00
C=1.2000000E+00

## SRq

Enables or disables service request and allows definition of the requirements for asserting SRq.

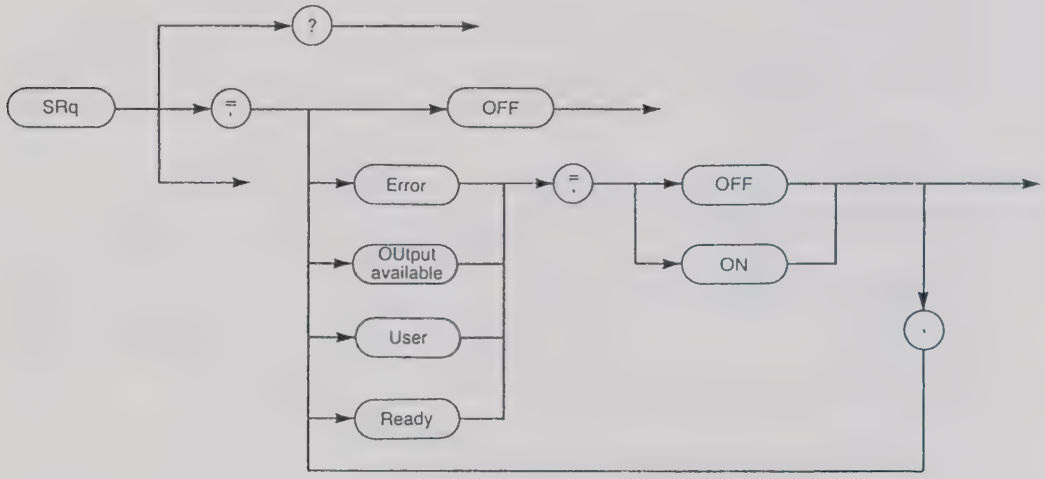



Fig 3.35 *SRQ Command*

The settings for the SRQ command can be described as follows:

| | |
|---|---|
| Error | – an SRQ is asserted on an error. Four types of error are handled: Command Interpretation, Command Execution, Calibration Fault, Input Message too long, i.e. >76 characters. |
| Output available | – an SRQ is asserted when the unit has output available. The output can be data, error messages, control information, etc. |
| User | – enables the voltmeter front panel SRQ control. |
| Ready | – an SRQ is asserted when the unit is not busy. SRQ will only be asserted when all of the instrument is ready. |

**Examples**

| | |
|---|---|
| SRq,User=ON | – enables the voltmeter front panel SRQ control. |
| SRq | – if SRq,User=ON, this command causes SRQ to be asserted with the user bit set. |
| SRq? | – produces a reply of the form:<br>SRq,Error=OFF,User=ON,<br>Output=OFF,Ready=OFF |

## STATistics

Enables selection and definition of the Statistics program

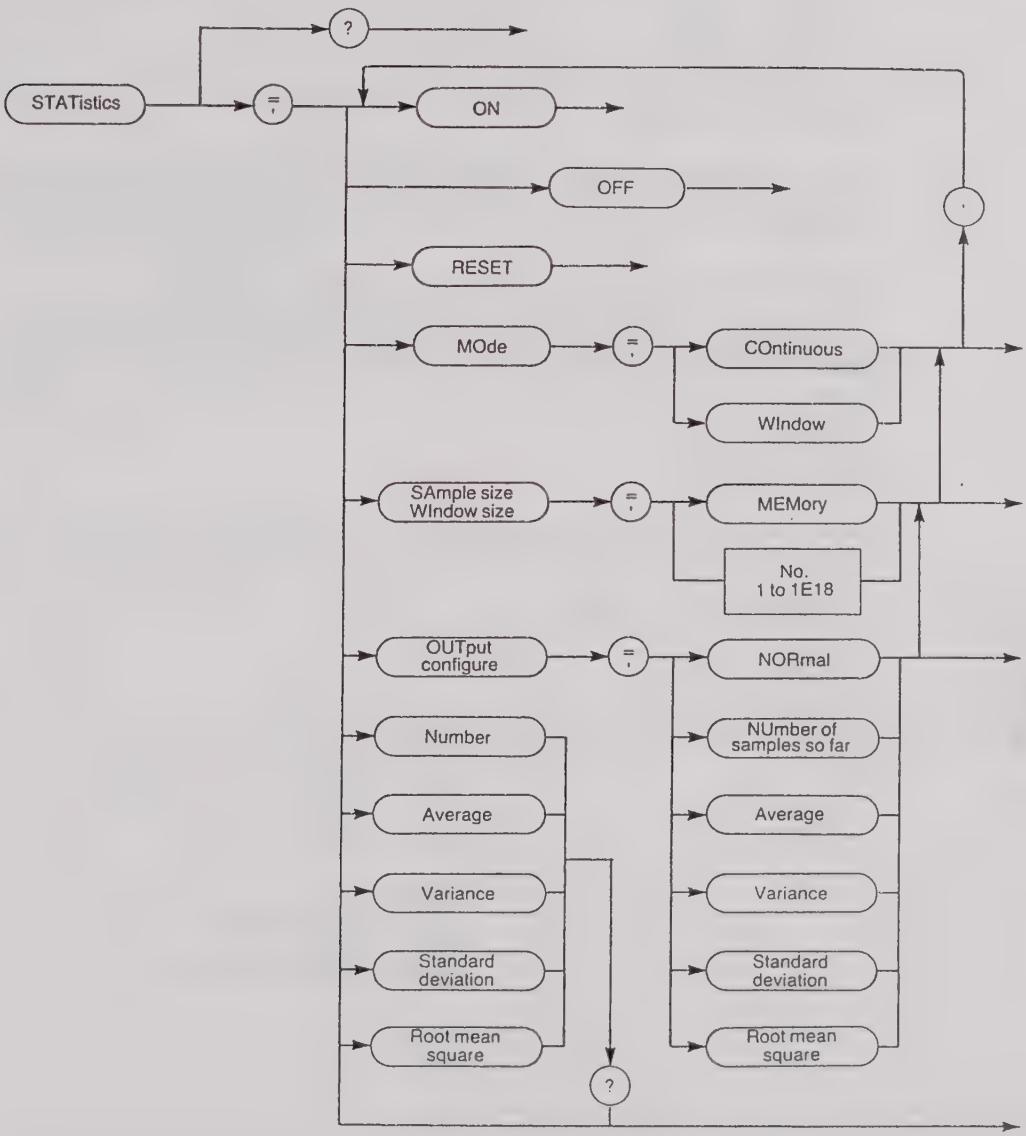



**Fig 3.36** *Statistics Command*

The Statistics command can be used to set up all of the program functions at one time, a particular function of the program or query a result.

The program has two modes of operation:

Continuous sampling
Window sampling

In continuous sampling, the running variables are constantly updated and an output result produced for every input.

In window sampling, the result is derived from a user defined sample size up to a maximum of 1E18. After the result has been produced, the variables are reset and a new sample collected.

Both modes may be configured to pass on any of the following outputs:

Normal (reading passed on unaltered)
Number so far (number of samples collected so far)
Average
Variance
Standard deviation
Root mean square

All of the outputs, with the exception of 'Normal', are stored in the unit's program memory and can be recalled at any time during the program's life by entering a query command, e.g.

STATistics,Root mean square?

**Examples**

STATistics,MOde=COntinuous,OUTput=Variance,On
– selects a mode of continuous sampling with a variance output and turns the program on.

STATistics?
– produces a reply of the form:

Statistics=OFF
Mode=Window Sampling
Output=Variance
Sample Size=10.0000000E+00

## STOp

Aborts all measurement processes, clears the output queue and also clears any unexecuted input commands.

**Fig 3.37** *Stop Command*

This command causes operation over the GP-IB to stop at the next character and the RS232 to stop at the next end of line.

If the Stop command is sent immediately after other commands to the voltmeter, some, if not all, of those commands may be ignored. This is because the commands are being aborted before they have been implemented. The Stop command should therefore never be used in a multicommand line.

### TEst

Causes execution of the internal test sequence.

**Fig 3.38** *Test Command*

## TIme

Used to set the voltmeter's real time clock.

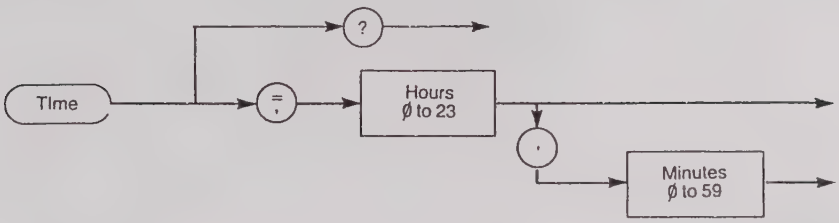



**Fig 3.39** *Time Command*

**Examples**

TIme=15,45 – sets the unit's clock to 3.45pm.

TIme? – produces a reply of the form:

Time = 15,45,30.4

## TRigger

Used to activate a function set up under ARM.

**Fig 3.40** *Trigger Command*

If no function is armed, TRigger has the same effect as MEAsure,SIngle

**Examples**

MEAsure,5,CHannel,1,5,9,ARM
– defines a series of 5 scans of the channel set specified.

TRigger – executes the series.

## 3 Command Execution Order

The order in which commands are executed does not necessarily follow the command sequence. This allows the user to change parameters such as the format of the output, turn outputs on or off, turn programs on or off, alter program constants, change the mode of the history storage, etc., even though, for example, a clock controlled measurement has begun.

In certain circumstances, operating a non-sequential system requires better user programming, e.g. in the program

```
MEAsure,15
Output,Gp-ib,ON
DUmp
```

the Output and history Dump commands are executed before all the results of the Measure command have been generated. To execute the Output and Dump commands after completion of the measurements requires some means of detecting the completion before sending the commands, e.g. SRQ on Ready.

### 3.1 Command Message Flow

Figure 3.41 shows the functional areas under which commands are grouped and how areas interact with each other.

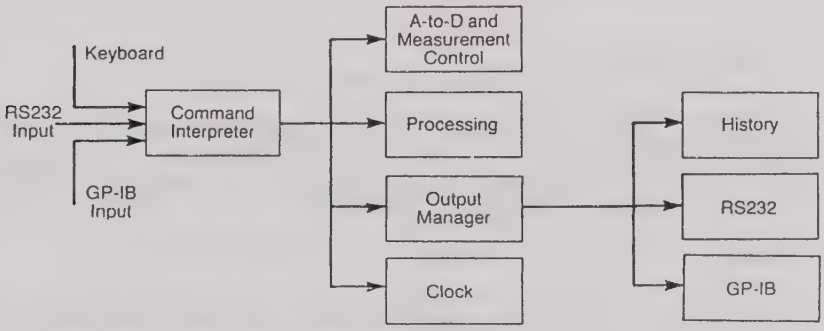



Fig 3.41 *Command Functional Areas*

Each functional area receives its own commands in sequence but commands to parallel areas may be executed in parallel. There are three exceptions:

1. The Test command puts the 7081 into a 'stopped' state to prevent further input until the self test is complete.

2. The Stop command is actioned almost immediately upon receipt (See page 3.44).

3. Only one Dump command can be actioned at any one time.

Table 3.5 indicates which commands are grouped under which functional areas

Table 3.5 *Command Groupings*

| Functional Area | Commands |
|---|---|
| A-to-D and Measurement Control | BEGin,CALIBRATE,CHannel,CLock,DELAy,DRift,ENd, INTerval,MEAsure,MODe,NInes,NUll,RANge,TRigger. |
| Processing | COmpute,DIGital filter,LImits,MEMory,RATio,SCale, STATistics. |
| Output Manager | BEEp,CAPitals lock,DELImit,DISplay,DUmp,ERror, FOrmat,HElp,HIstory,INItialise,LOck front panel,Output, Pad count,SRq. |
| Clock | DAte,TIme. |
| History via Output Manager | DUmp,HIstory. |
| RS232 via Output Manager | DELImit,Pad count. |
| GP-IB via Output Manager | SRq. |

## 4 Error Messages

7081 error messages are output in different forms depending upon the interface used and the command settings of that interface. Provided the display is on, error messages 20 to 50 always appear when they arise. When the display is set to monitor, the first 20 characters of the verbose form of a command are displayed.

### 4.1 Display

If the voltmeter display is set to Monitor, verbose error messages will always appear. Error numbers never appear in the display but messages 20 to 50 will always appear in their verbose form regardless of the display setting.

### 4.2 RS232

The RS232 interface can be enabled for either brief or verbose error messages. If brief messages are enabled, the error numbers (E numbers) will appear but messages 20 to 49 will always appear in their verbose form, provided the interface is on, regardless of the error setting.

## 4.3 GP-IB

No error messages will be output, if error is set to brief, but the serial poll byte abnormality code will be set to a value between 00 and 11 depending upon the type of message generated, i.e.

| | |
|---|---|
| Command Interpretation Error | Code=00 |
| Command Execution Error | Code=01 |
| Calibration Fault | Code=10 |
| Input Message Too Long (more than 76 chars) | Code=11 |

The GP-IB will receive verbose messages, if error is set to verbose.

## 4.4 Message Groups

The messages can be divided up into the following four groups.

Error messages 0 to 19 may be displayed in response to a command input and are grouped under the GP-IB abnormality code settings shown in Table 3.6.

**Table 3.6** *Errors 0 to 19*

| Brief Message | Verbose Message | GP-IB Abnormality Code |
|---|---|---|
| OK | Command Syntax OK | — |
| E1 | Command Incomplete Before Char No. This Part: | 00 |
| E2 | Numeric Not Expected Before Char No. This Part: | 00 |
| E3 | 'Word' Unrecognised Before Char No. This Part: | 00 |
| E4 | Invalid Separator Before Char No. This Part: | 00 |
| E5 | Numeric Out of Range Before Char No. This Part: | 00 |
| E6 | Too many Arguments Before Char No. This Part: | 00 |
| E7 | Argument Missing Before Char No. | 00 |
| E8-9 | Reserved For Future Use | — |
| E10 | Record nnnn Not Present (result of DUmp) | 01 |
| E11 | Null Mode Illegal | 01 |
| E12-19 | Reserved For Future Use | — |

Error messages 20 to 49 may be displayed in response to a Calibration, Dump, Compute or Null command, or an Initialise or Power-up sequence. Refer to Table 3.7 for a listing of the messages.

**Table 3.7** *Errors 20 to 49*

| Brief Message | Verbose Message | GP-IB Abnormality Code |
|---|---|---|
| E20 | Null Too High | 01 |
| E21 | Complete (null sequence) | — |
| E22 | *Initialised* | — |
| E23 | Cal OK | - |
| E24 | NVM Fail | 10 |
| E25 | Dump Fail | 10 |
| E26 | Cal Incomplete | 10 |
| E27 | *Resumed* | — |
| E28 | Cal Ref Out Of Range | 10 |
| E29 | Zero Too Large | 10 |
| E30 | Vref Out Of Range | 10 |
| E31 | Factor Out Of Range | 10 |
| E32 | Cal Not Enabled | 10 |
| E33 | Pass (refers to self test) | — |
| E34 | Fail, Test 0Vdc | 01 |
| E35 | Fail, Test 10Vdc | 01 |
| E36 | Fail, Test Ohms | 01 |
| E37 | Fail, Test AC | 01 |
| E38 | Insufficient History | 01 |
| E39 | Complete (refers to COmpute,HIstory) | — |
| E40 | No Programs On | 01 |
| E41-49 | Reserved For Future Use | — |

Table 3.8 *Errors 50 to 59*

| Brief Message | Verbose Message | GP-IB Abnormality Code |
|---|---|---|
| E50 | No History Present | 01 |
| E51-59 | Reserved For Future Use | — |

Table 3.9 *Errors 60 Onwards*

| Brief Message | Verbose Message | GP-IB Abnormality Code |
|---|---|---|
| E60 | Complete (refers to DUmp command) | — |
| E61 onwards | Reserved For Future Use | — |

## 4.5 Examples

The following examples show both the brief and verbose responses.

| | | |
|---|---|---|
| DUmp | OK | Command Syntax OK |
| | E50 | No History Present |
| DUmp,1,To,5 | OK | Command Syntax OK |
| | 5.97425 | 5.97425 |
| | 5.97086 | 5.97086 |
| | 5.89322 | 5.89322 |
| | E10 | Record 4 Not Present |
| | E10 | Record 5 Not Present |
| | Complete | Complete |
| MEAsure,CHannel,1,To | E1 | Command Incomplete Before Char No. 22<br>This Part: To |
| MODe=1 | E2 | Numeric Not Expected Before Char No. 8<br>This Part: 1 |
| Filter,ON | E3 | 'Word' Unrecognised Before Char No. 7<br>This Part: Filter |
| SCale,M=2=C=4 | E4 | Invalid Separator Before Char No. 10<br>This Part: 2= |

# Chapter 4
# RS232 Operation

**Contents**


| | | *Page No.* |
|---|---|---|
| 1 | Introduction | 4.2 |
| 2 | Hardware and Terminology | 4.2 |
| 3 | Transmission Speeds | 4.2 |
| 4 | Command Language | 4.3 |
| 5 | Input and Output | 4.3 |
| 6 | Programming Examples | 4.4 |

This chapter describes the use of 7081 via the RS232 interface.

## 1 Introduction

The 7081 voltmeter can be connected to any equipment conforming to the RS232 Standard via its RS232 Interface. This chapter details the interface and the voltmeter's RS232 operation.

## 2 Hardware and Terminology

In RS232 terms, 7081 is designated the Data Communication Equipment (DCE) and its associated terminal the Data Terminal Equipment (DTE). Input commands from the DTE are referred to as Received Data, whilst output signals to the DTE are Transmitted Data. The data is carried, in serial bytes comprising serial bits, on two signal wires.

There are four handshake wires:

| | | |
|---|---|---|
| Data Terminal Ready | – | the DTE is ready to send primary data. |
| Data Set Ready | – | the DCE is ready to process input data. |
| Clear to Send | – | the DCE is ready to send data. |
| Carrier Detect | – | the DCE acknowledges receipt of the DTE's signal carrier. |

The remaining two lines used are Protective Ground (supply 0V) and Common Return (signal 0V).

## 3 Transmission Speeds

Most RS232 devices can operate at up to 9600 baud and the 7081 speed must be set to match that of the terminal device. This is achieved via three switches mounted on printed circuit board 3. The possible switch settings are given in Table 4.1. Note that the normal factory setting is 300 baud

**Table 4.1** *Baud Rate Switches*

| Baud Rate | Switch 1 | Switch 2 | Switch 4 |
|---|---|---|---|
| 110 | ON | ON | ON |
| 150 | OFF | ON | ON |
| 300 | ON | OFF | ON |
| 600 | OFF | OFF | ON |
| 1200 | ON | ON | OFF |
| 2400 | OFF | ON | OFF |
| 4800 | ON | OFF | OFF |
| 9600 | OFF | OFF | OFF |

## 4 Command Language

The command language used for 7081, over the RS232 link, is the same as that used over the GP-IB (see previous chapter for language details) except that, when operating over an RS232 link, the language is device independent and can be simply keyed into the terminal device.

If, while keying in a command message string, the operator makes an error, the mistake is easily rectified by the use of the terminal DELETE or RUB OUT key.

It is possible to turn off the echo to an RS232 terminal, from the voltmeter, by typing CTRL N. To turn the echo back on again enter CTRL O. The echo to the terminal takes account of the Capitals Lock Command, i.e., with CAPitals lock=ON, all the echoed commands will be in capitals with the commas and colons replaced by spaces.

## 5 Input and Output

If an associated terminal/computer attempts to input messages to the 7081 at a faster rate than the unit can implement them, 7081 will ignore some of the commands. The voltmeter can queue up to a maximum of three messages before it starts to ignore a percentage of the input.

Owing to the lack of handshake, RS232 output messages may be lost if the terminal/computer is not ready to receive. The unit will still accept messages if the output is queued.

Note: A message is defined, in this context, as an input line terminated by a Carriage Return, Line Feed or EOI.

The following commands are relevant to output control.

1. OUTPUT — allows generated data to be routed only to those interfaces desired. If both RS232 and GP-IB are on, output occurs at the rate of the slowest device to ensure that both outputs remain synchronised.

2. FORMAT — controls the form of the measurement results output.

3. DELIMIT — defines the 'end of line' character.

4. ERROR — if the RS232 output is on and ERror = Verbose, error reporting messages will also appear at the output. ERror = Brief will result in error numbers.

5. PAD COUNT — defines the number of nulls inserted before the 'end of line' character

These commands enable powerful editing of information transmitted to the output and into the history file.

# 6 Programming Examples

The following examples show how the 7081 facilities may be used via RS232 remote control. In all the examples it is assumed that the voltmeter is in the initialised state.

## 6.1 Measurement Examples

Having switched the voltmeter and associated devices on, the following may be entered.

**Example 1**

```
Test
Output, RS232, On
Error = Verbose
Format, DVM, Expanded
Mode = TR
Range = Auto
Nines = 6
Measure, Single
```

This example takes the unit through a self test and then sets the RS232 Interface output on with verbose error messages. The format of the output display is set to DVM, Expanded. A single true resistance measurement is initiated, with automatic range and scale length of 6 × 9, and the result displayed at the interface output.

**Example 2**

```
Mode = VDC
Nines = 5
Clock, Elapsed
Begin = 0, 0, 30
Interval = 0, 1
End = 0, 3
Measure, Clock Controlled
```

Example 2 changes the measurement function to Vdc and the scale length to 5 × 9. The system clock is then set up and three clock controlled measurements are taken and displayed at the interface output.

**Example 3**

```
Mode = VAC
Nines = 7
Range = 100
Format = Engineering
Measure, Continuous
_
_
_
_
Stop
```

In this case, the mode is set to Vac, the scale length to 7 × 9, the range to 100 and the output format to Engineering, Expanded. Continuous measurements are taken and displayed until the Stop command is entered

## 6.2 Processing Examples

Assuming that the voltmeter has been switched on, tested and the interface set up for verbose error messages, the following examples may be entered.

**Example 1**

```
Format, DVM, Compressed
Ratio, Mode = Main/N, N = 11
Ratio, On
Measure,2
–
–
–
–
Ratio, Off
```

Example 1 sets up the Ratio program to Main/N with a value of 11 for the constant N. The program is turned on and two measurements are taken and processed. The processed results are then output in DVM, Compressed format and the program turned off.

**Example 2**

```
Scale, M = 6, C = 4, On
Nines = 5
Measure, 10
–
–
–
–
–
–

Statistics, Mode = Window, Window Size = 10, Output = Average, On
Measure, 10
```

The Scale program is selected and defined, in example 2, and ten measurments taken. The processed results are displayed. The Statistics program is then enabled and defined, ten measurements taken and the average result displayed. As the Scale program has not been turned off, the measurements are processed by both programs.

**Example 3**

```
Limits, Mode = Window, Sample Size = 13, Output = Peak to Peak, On
Format = DVM, Expanded
Measure, 13
```

Example 3 sets up and defines the Limits program. Thirteen measurements are taken and the processed result displayed in DVM, Expanded format.

**Example 4**

```
Digital Filter, Mode = Walking Window, Window Size = 12, On
Measure, 23
```

This example sets up the Digital Filter program and initiates 23 readings. The program window is first filled and then 12 results are displayed.

## 6.3 Overload

If an overload occurs and the voltmeter interface output is set for Expanded format, an overload message will appear in the output.

Example 1 shows such an occurrence. The voltmeter replies are shown in parenthesis.

**Note:** The output errors are set to Brief.

**Example 1**

```
Output,RS232,On
(OK)
Error=Brief
(OK)
Format=DVM,Expanded
(OK)
Mode=VAC+VDC+Filter
(OK)
Range=0.1
(OK)
Nines=5
(OK)
Measure,3
(OK)
(0.28893        Overload Time = 11,22,11.1,Day=01)
(0.28893        Overload Time = 11,22,13.3,Day=01)
(0.28892        Overload Time = 11,22,15.4,Day=01)
```

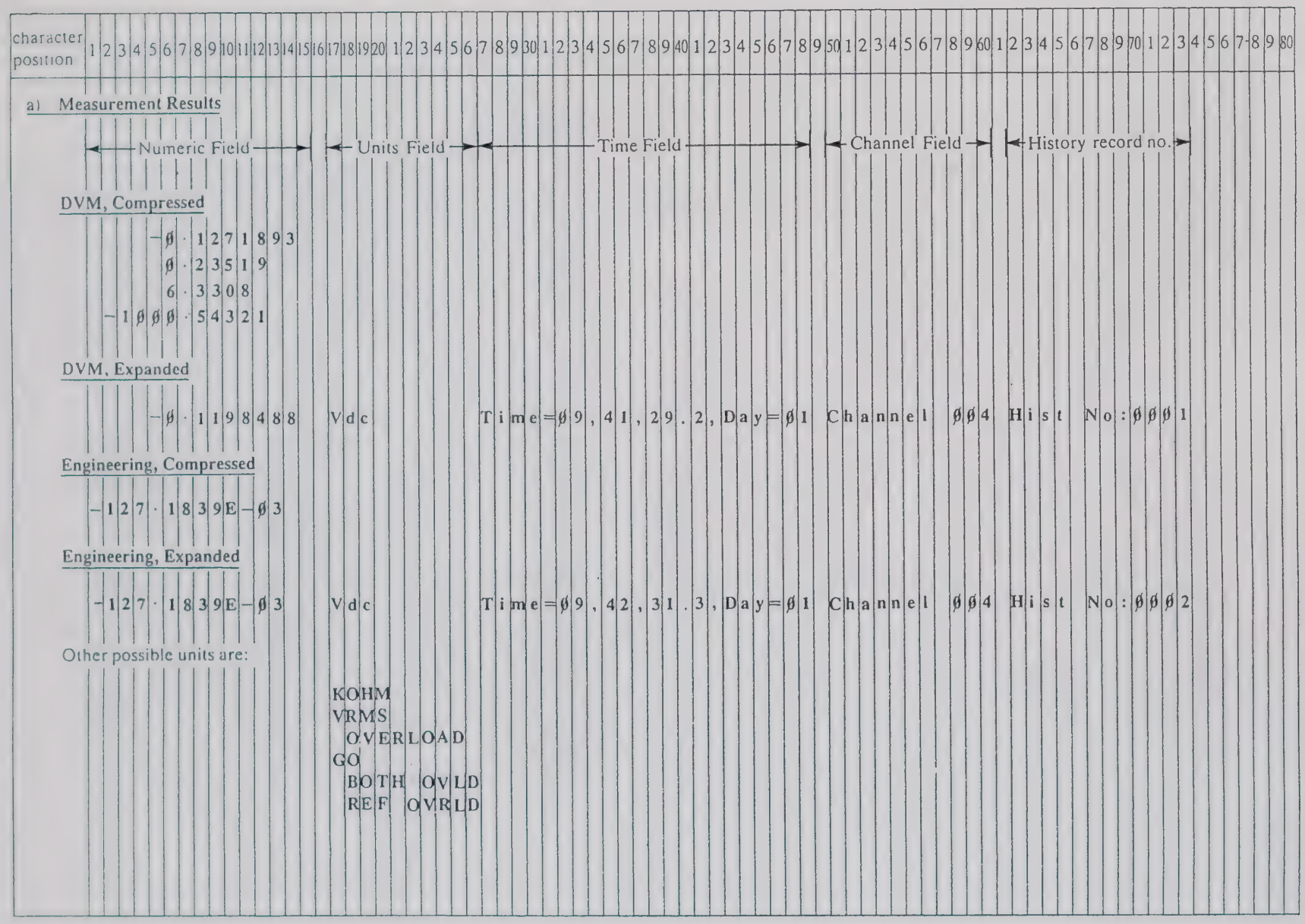
## 6.4 History Dump

When a Dump command is entered into a voltmeter with format set to expanded, the history file number for each result will appear in the output. Any requested file numbers that are not available will be indicated by an error message. Refer to Example 1. The voltmeter replies are again shown in parenthesis.

**Example 1**

```
Format.Engineering.Expanded
(Command Syntax OK)
Dump.13.To.30
(Command Syntax OK)
(11.00E+00 Go Time=12,15,12.7,Day=03 Channel 001 Hist No:0013)
(12.00E+00 Go Time=12,15,13.4,Day=03 Channel 001 Hist No:0014)
(13.00E+00 Go Time=12,15,14.1,Day=03 Channel 001 Hist No:0015)
(14.00E+00 Go Time=12,15,14.8,Day=03 Channel 001 Hist No:0016)
(15.00E+00 Go Time=12,15,15.5,Day=03 Channel 001 Hist No:0017)
(16.00E+00 Go Time=12,15,16.3,Day=03 Channel 001 Hist No:0018)
(17.00E+00 Go Time=12,15,17.0,Day=03 Channel 001 Hist No:0019)
(18.00E+00 Go Time=12,15,17.7,Day=03 Channel 001 Hist No:0020)
(19.00E+00 Go Time=12,15,18.4,Day=03 Channel 001 Hist No:0021)
(20.00E+00 Go Time=12,15,19.1,Day=03 Channel 001 Hist No:0022)
(21.00E+00 Go Time=12,15,19.8,Day=03 Channel 001 Hist No:0023)
(22.00E+00 Go Time=12,15,21.2,Day=03 Channel 001 Hist No:0024)
(15.00E+00 Go Time=12,15,15.5,Day=03 Channel 001 Hist No:0025)
(Record 26 Not Present)
(Record 27 Not Present)
(Record 28 Not Present)
(Record 30 Not Present)
(Complete)
```

# Appendix 1
# Examples of Output Format


character position
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 1 2 3 4 5 6 7 8 9 30 1 2 3 4 5 6 7 8 9 40 1 2 3 4 5 6 7 8 9 50 1 2 3 4 5 6 7 8 9 60 1 2 3 4 5 6 7 8 9 70 1 2 3 4 5 6 7 8 9 80
a) Measurement Results
Numeric Field
Units Field
Time Field
Channel Field
History record no.
DVM, Compressed
−ø·127893
ø·23519
6·3308
−1øøø·54321
DVM, Expanded
−ø·1198488 Vdc Time=ø9,41,29.2,Day=ø1 Channel øø4 Hist No:øøø1
Engineering, Compressed
−127·1839E−ø3
Engineering, Expanded
−127·1839E−ø3 Vdc Time=ø9,42,31.3,Day=ø1 Channel øø4 Hist No:øøø2
Other possible units are:
KOHM
VRMS
OVERLOAD
GO
BOTH OVLD
REF OVRLD

Notes: 1/ In DVM format, the decimal point occupies a fixed position and the number therefore gives a positional indication of magnitude.

2/ In Engineering format, the number occupies a fixed position with constant precision for the integration time selected. The exponent, which is always shown, is allowed to change in intervals of three, the decimal point may thus occupy one of three position.

3/ If an overload is present, the overload indication replaces the units field.

4/ The channel field is left blank if channel = ØØØ is selected (normal voltmeter terminals).

5/ In MEAsure, CONtinuous the time field contains the time of completion of each result, other measurement modes record the start time of the measurement. An elapsed time count of the number of days since either the instrument was switched on, or the current clock controlled sequence began is also present. The day of switch on is counted as day Ø1, the first day of clock controlled measurement is day ØØ.

6/ The history record number is appended to dumped history records when FORmat = EXpanded. If the history file is stored in compressed format, the dumped results are in engineering format. If FORmat = Expanded but HISTory = COMpressed, the units and channel field would be blank and the time values all zero.

character position 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 1 2 3 4 5 6 7 8 9 30 1 2 3 4 5 6 7 8 9 40 1 2 3 4 5 6 7 8 9 50 1 2 3 4 5 6 7 8 9 60 1 2 3 4 5 6 7 8 9 70 1 2 3 4 5 6 7 8 9 80

b) Other Outputs

Reply to: Limits, peak to peak?

```
P to P =  9 4 · 2 9 8 7 2 Ø 2 E − Ø 3
```

Reply to: Limits, min?

```
M i n  =  − 1 . 2 7 2 8 Ø 2 6 1 E + Ø Ø
```

Reply to: Begin?

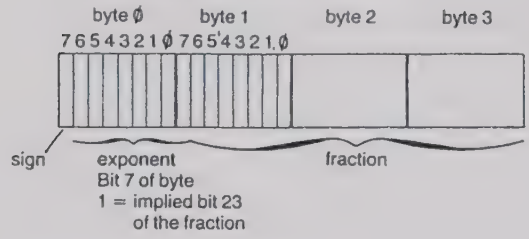
```
B e g i n  =  Ø Ø , Ø Ø , 1 Ø . Ø , D a y = Ø Ø
```

Notes: 1/ Program result replies always have the form text/space/=space/numeric where the number value is in engineering format with a minus sign if the value is negative.

2/ Begin, end and interval times use a fixed numeric format, i.e. text/space/=/space/HH, MM, SS.S, Day = DD.

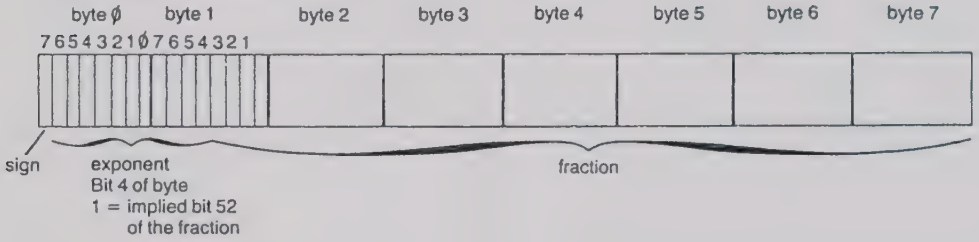
## Binary Compressed

For 3 to 7 nines, single precision is used =




The exponent bias is 127, therefore the number represented is fraction x 2 (exponent − 127)

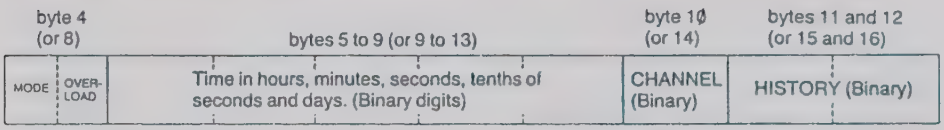
For 8 nines, double precision is used =




The exponent bias is 1Ø23, therefore the number represented is fraction x 2 (exponent − 1Ø23)

## Binary Expanded

If FOrmat = Binary, EXpanded is selected, a binary equivalent of the ASCII expanded information is added to the result string:




Mode nibble:
- Ø = Vdc
- 1 = Vac
- 2 = Kohm
- 3 = Vac + dc
- 4 = test (Ø Vdc)
- 5 = test (1Ø Vdc)
- 6 = test (Kohm)
- 7 = test (Vac)
- 8 = ref. terminals only
- 9 = Vac + filter
- 1Ø = Vac + dc + filter
- 11 = true Kohm
- 12 = kVdc probe

Overload nibble:
- Ø = no overload
- 1 = main terminals overload
- 8 = reference terminals overload
- 9 = both terminals overload

No other values are used.

# Appendix 2
# Adopted Settings

At power-up (resumed), initialise or Device Clear, the voltmeter adopts known states for its command settings. Table A 2.1 gives a list of these states.

**Table A 2.1** *7081 Adopted Settings*

| ITEM | INITIALISED | RESUMED | DEVICE CLEARED |
|---|---|---|---|
| **Keyboard** | | | |
| Queued Messages | deleted | deleted | deleted |
| Local | local | local | Not changed (unless REN unasserted) |
| Lock | off | Not changed | Not changed |
| **GP-IB Input** | | | |
| Queued messages | deleted | deleted | deleted |
| **RS232 Input** | | | |
| Queued messages | deleted | deleted | deleted |
| Echo | on | Not changed | Not changed |
| **Clock** | | | |
| Time | Not changed | Not changed | Not changed |
| Date | Not changed | Not changed | Not changed |
| Year | Not changed | Not changed | Not changed |
| **A-to-D** | | | |
| Nines | 6 | 6 | 6 |
| Channel | 0 | 0 | 0 |
| Pull-in Delay | 2 | 2 | 2 |
| Drop-out Delay | 2 | 2 | 2 |
| Trigger Delay | Normal | Normal | Normal |
| Range | 1000 Auto | 1000 Auto | 1000 Auto |
| Drift Correct | Auto | Auto | Auto |
| Mode | Vdc | Vdc | Vdc |
| Arm | off | off | off |
| Null | off | off | off |
| Clock Mode | real | Not changed | Not changed |
| Begin | zeroed | Not changed | Not changed |
| Interval | zeroed | Not changed | Not changed |
| End | zeroed | Not changed | Not changed |
| Time control | off | off | off |

Table A 2.1 *7081 Adopted Settings*

| ITEM | INITIALISED | RESUMED | DEVICE CLEARED |
|---|---|---|---|
| **Processing** | | | |
| Compute | off | off | off |
| Ratio | off | off | off |
| Ratio Mode | Main/Ref | Not changed | Not changed |
| Ratio Constants | zeroed | Not changed | Not changed |
| Dig filter | off | off | off |
| Dig filter mode | Walking Window | Not changed | Not changed |
| Dig filter size | 10 | Not changed | Not changed |
| Scale | off | off | off |
| Scale constants | zeroed | Not changed | Not changed |
| Stats | off | off | off |
| Stats mode | continuous | Not changed | Not changed |
| Stats sample size | 10 | Not changed | Not changed |
| Stats output | normal | Not changed | Not changed |
| Stats results | reset | reset | reset |
| Limits | off | off | off |
| Limits mode | continuous | Not changed | Not changed |
| Limits sample size | 10 | Not changed | Not changed |
| Limits high limit | +1.9E18 | Not changed | Not changed |
| Limits low limit | −1.9E18 | Not changed | Not changed |
| Limits output | normal | Not changed | Not changed |
| Limits results | reset | reset | reset |
| **Output Manager** | | | |
| Error | brief | Not changed | Not changed |
| Format | compressed, dvm | Not changed | Not changed |
| **History File** | | | |
| Format | compressed, 1500 | Not changed | Not changed |
| Mode | roll around | Not changed | Not changed |
| Direction | forward | Not changed | Not changed |
| Contents | empty | Not changed | Not changed |
| **Display** | | | |
| Queued messages | deleted | deleted | deleted |
| On/Off | on | Not changed | Not changed |
| **GP-IB Output** | | | |
| Queued messages | deleted | deleted | deleted |
| On/Off | off | off | off |
| SRQ enables | off | Not changed | Not changed |
| **RS232 Output** | | | |
| Queued messages | deleted | deleted | deleted |
| On/Off | off | Not changed | Not changed |

# Index


Adapted Settings, A2.1

Combined Systems, 1.2
Commanding Measurements, GPIB, 2.2
Command Input and Output, 2.7
Command Language, General, 3.2
Command Language, Symbol Definition, 3.3
Command Words, General, 3.3
Command, BEEp, 3.4
Command, BEGin, 3.5
Command, CALIBRATE, 3.6
Command, CAPitals Lock, 3.7
Command, CHannel, 2.4, 3.8
Command, CLock, 2.4, 3.9
Command, COmpute, 3.10
Command, DAte, 3.11
Command, DELay, 3.12
Command, DELImit, 3.13
Command, DIGital Filter, 3.14
Command, DISplay, 3.16
Command, DRift, 3.17
Command, DUmp, 3.18
Command, ENd, 3.19
Command, ERror, 3.20
Command, Execution Order, 3.49
Command, FOrmat, 3.21
Command, Groupings, 3.50
Command, HElp, 3.23
Command, HIstory, 3.24
Command, INItialise, 3.25
Command, INTerval, 3.26
Command, LImits, 3.27
Command, LOck( out) Front Panel, 2.9, 3.30
Command, MEAsure, 2.3, 3.31
Command, MEMory, 3.33
Command, Message Flow, 3.49
Command, MODe, 3.34
Command, NIne, 3.35
Command, NUll, 3.36
Command, Output, 3.37
Command, Pad Count, 3.38
Command, RANge, 3.39
Command, RATio, 3.40
Command, SCale, 3.41
Command, SRq, 3.42
Command, STATistics, 3.43
Command, STOp, 3.45
Command, TEst, 3.46
Command, TIme, 3.47
Command, TRigger, 2.5, 3.48
Compute Off, 2.10

Diagram, Symbol Syntax, 3.3

Error Indication, Serial Poll, 2.9
Error Messages, 0 to 19, 3.51
Error Messages, 20 to 49, 3.52
Error Messages, 50 onwards, 3.53
Error Messages, Display, 3.50
Error Messages, Examples, 3.53

Front Panel, Lock Out, 2.9
Functions, GPIB, 2.7

GPIB Bus Structure, 1.4
GPIB Handshaking, 1.5
GPIB, 3.50
GPIB, Cabling, 1.3
GPIB, Commanding Remote and Local, 2.2
GPIB, Command Language, General, 1.6
GPIB, Devices, 1.3
GPIB, General, 1.3, 2.2
GPIB, Management, 1.4
GPIB, Measurement Command, General, 2.2
GPIB, SRQ, 1.5
GPIB, Switch Selection, 1.6

History Dump, 2.10

IEEE 488/1978 see GPIB, 1.5
Input and Output, Commands, 2.7
Interface, GBIB, 1.3
Interface, RS232, 1.2
Interfaces, General, 1.2
Interrogation Characters, 2.9
Invalid Characters, (GPIB), 2.6

Local, GPIB, 2.2

Message Groups, 3.51
Message Protocols, (GPIB), 2.6
Messages, Error, 0 to 19, 3.51
Messages, Error, 20 to 49, 3.52
Messages, Error, 50 onwards, 3.53
Messages, Error, Display, 3.50
Messages, Error, Examples, 3.53
Mode, Interrogation Character, 2.9

Output Format Examples, A1.2, 1.3
Output Format Examples, Binary Compressed, A1.4
Output Format Examples, Binary Expanded, A1.4

Parallel Poll Sharing, GPIB, 2.8
Parallel Poll, GPIB, 2.7

Parallel Poll Sharing, GPIB, 2.8
Parallel Poll, GPIB, 2.7
Processed Measurements, 2.10
Program Exit, 2.10
Programming Examples,
Clock Controlled Measurements, 2.14
Ratio Program, 2.11
Scale and Limits Program, 2.13
Single Measurement, 2.10
Statistics Program, 2.12

Range, Interrogation Character, 2.9
Remote, GPIB, 2.2
RS232C, Pin Assignments, 1.3
RS232, 1.2, 3.50
RS232, Baud Rate Switches, 4.2
RS232, Command Language RS232,
Input Output, 4.3
RS232, GPIB, 3.50
RS232, General, 1.3, 4.2
RS232, History Dump, 4.6
RS232, Overload, 4.6
RS232, Programming Examples,
Measurement, 4.4
Processing, 4.5
RS232 Transmission Speeds, 4.2

Serial Poll Byte, 2.8
Serial Poll Enable, 2.8
Serial Poll, Error Indicator, 2.9
SRQ, General (GPIB), 1.5
SRQ, GPIB, 2.9
Switch Selection, (GPIB), 1.6

MP MASTER
MA20080